



UNIVERSIDAD COMPLUTENSE
MADRID

TRABAJO FIN DE GRADO DEL GRADO EN INGENIERÍA INFORMÁTICA

Herramienta web de autodefensa digital.

Curso académico 2017/2018

Autor: Álvaro Asenjo Torrico
Director: Marcos Sánchez Elez

ÍNDICE

Palabras clave	6
Resumen	6
Summary	6
Introducción	7
Motivación	7
Definición del proyecto	7
KapiSolution	9
Objetivos	9
Formación acerca de la autodefensa digital	9
Herramienta colaborativa	9
Plataformas similares	12
Wikipedia	12
Guía de autodefensa catalana	13
Security Education Companion	13
Comparativa entre aplicaciones	14
Tecnologías similares	14
Metodología y plan de trabajo	15
Metodología de trabajo	15
Arquitectura	17
Control de versiones (Git y GitHub)	20
Especificación de requisitos	21
Requisitos funcionales	21
Requisitos no funcionales	23
Restricciones sobre perfiles profesionales y nivel de usuario	24
Tecnologías utilizadas	24
TinyMCE	27
Bootstrap	28

Funcionalidad de la herramienta	30
Usuarios no logueados	32
Registro	33
Inicio de sesión	35
Usuarios logueados	36
Generación de nuevo contenido	37
Modificación de contenido	38
Votaciones sobre contribuciones de usuarios	43
Pruebas de nivel	47
Perfil del usuario	50
Búsqueda	53
Evolución del proyecto	55
Conclusiones	56
Trabajo futuro	57
Enlaces y Bibliografía	57
Enlaces	57
Bibliografía	58
Wikipedia	58
Guía de autodefensa catalana	58
Security Education Companion	58
Bootstrap	58
Git	58
Apéndice	59
Casos de uso	59

ÍNDICE DE FIGURAS Y TABLAS

Figura 1. Formas de obtener nivel en la herramienta	9
Figura 2. Formas de colaborar según tipo de usuario	11
Tabla 1. Comparativa entre herramientas similares	15
Figura 3. Metodología de trabajo para el control de versiones	18
Figura 4. Arquitectura de una aplicación bajo el stack LAMP	18
Figura 5. Arquitectura de trabajo durante el desarrollo de la aplicación	20
Figura 6. Estructura de la base de datos	21
Figura 7. Sistema de control de versiones distribuido.	22
Figura 8. Ejemplo de uso de TinyMCE	30
Figura 9. Comparativa vista de móvil con ordenador	32
Figura 10. Flujo de actividades de la aplicación para usuarios no registrados	33
Figura 11. Flujo de actividades de la aplicación para usuarios registrados	34
Figura 12. Aspecto de la herramienta para un usuario no logueado.	35
Figura 13. Aspecto de la herramienta para un usuario logueado.	40
Figura 14. Casos de uso de la aplicación	62

Agradecimientos

En primer lugar, me gustaría agradecer a mis padres, Juan y Rosa, la confianza que siempre tuvieron en mí para finalizar este grado. Por su apoyo y su comprensión en los momentos más complicados de mi carrera universitaria.

Agradecer también al director de este proyecto Marcos Sánchez Élez, que desde el primer día que me propuso este trabajo final de grado, ha estado disponible para reunirse conmigo y darme apoyo en todo lo que he necesitado.

Por último, no me quiero olvidar tampoco de aquellos compañeros que me han echado una mano cuando se lo he pedido, y a aquellos que sin pedírselo han estado apoyándome durante este camino. A Sergio, Víctor, Juan, muchas gracias.

Palabras clave

Aplicación web; Artículos; Seguridad; Autodefensa; Herramienta colaborativa; Comunidad; Formación

Resumen

El proyecto tiene como objetivo la implementación de una herramienta de autodefensa digital, donde se pretende que un usuario, tanto si dispone de cierto nivel como si comienza su formación desde cero, pueda formarse en seguridad. Además la herramienta dispone de artículos dedicados a perfiles profesionales, por lo que se pretende que la aplicación sirva como consulta a la hora de desarrollar una actividad laboral que se relacione de alguna manera con el uso de la red de una forma segura, por ejemplo, se ha desarrollado el perfil de informático, periodista y abogado, con artículos específicos para estos perfiles profesionales. Se ha desarrollado un sitio web donde los usuarios pueden consultar artículos relacionados con la autodefensa digital en sus sectores profesionales, o simplemente por el hecho de mejorar en este ámbito. La herramienta se ha diseñado pensando en un entorno de co-creación, donde aquel usuario con permisos suficientes es capaz de generar contenido y fomentar la idea de que la herramienta crezca en torno al conocimiento. También permite al usuario ir avanzando en su nivel de autodefensa superando diferentes pruebas de nivel que a su vez le otorgarán los permisos necesarios para colaborar en la aplicación.

Summary

The project aims to implement a digital self-defense tool where users with or without level can train in security, in addition the tool has articles dedicated to professional profiles, so it is intended that the application serves as a place to find the information at the time of developing a work activity that is related in some way to the use of the web. This is, a website where users can consult articles related to digital self-defense in their professional sectors, or simply for the fact of improving in this area. The tool includes different profiles (computer, lawyer, journalist) in which you can find articles related to self-defense for each profile. The tool has been designed with a co-creation environment in mind, where the user with sufficient permissions is able to generate content and encourage the idea that the tool grows around knowledge. It also allows the user to advance in their level of self-defense by passing different level tests that in turn will grant the necessary permissions to collaborate in the application.

1. Introducción

1.1. Motivación

El objetivo era realizar una aplicación que reuniera una gran cantidad de conocimiento sobre autodefensa digital y que fuera posible encontrar artículos relacionados sobre este ámbito, permitiendo al usuario la búsqueda de información en una aplicación de fácil uso. Además queríamos otorgar a nuestra herramienta de un carácter colaborativo, donde cada usuario pudiera, bajo una serie de condiciones, generar contenido.

Las preguntas que nos hicimos antes de desarrollar nuestra aplicación fueron:

- **¿En qué lugar nos encontramos?**
 - Es por todos sabido que internet reúne una gran cantidad de información, pero la tarea de encontrar aquello que necesitamos suele ser en ocasiones algo árdua, tanto por la cantidad de información unas veces, como por el nivel de conocimientos previos necesarios otras veces. Por ello, y tras no encontrar un sitio web de estas características, comenzamos a trabajar sobre esta idea.
- **¿Existe una idea o aplicación similar a la nuestra, o bien existe ya algo desde lo que partir y mejorar?**
 - Como se ha mencionado, Internet es un amplio lugar de búsqueda y reunión de conocimiento, donde cualquier usuario con una base previa de conocimiento puede formarse en distintos ámbitos. Sin embargo, en el ámbito de la autodefensa es difícil encontrar un sitio web donde se reúna información en este ámbito y donde poder progresar en nuestros conocimientos. Durante la búsqueda de diferentes aplicaciones que estuvieran desarrolladas sobre nuestra misma idea, se encontraron una serie de plataformas similares que se exponen más adelante.

1.2. Definición del proyecto

Este proyecto tiene como objetivo la creación de un sitio web de consulta y formación en el ámbito de la autodefensa digital. Se pretende dotar al usuario de una herramienta web de fácil uso, donde pueda buscar diferentes artículos bajo el mismo ámbito. Durante la definición del proyecto, la idea principal era reunir en un sitio web información acerca de este ámbito, ya que la búsqueda no siempre es fácil con respecto a este tema, y en el caso de que el usuario encuentre lo que quiere, puede carecer de los conocimientos adecuados para entender los resultados de su búsqueda.

Imaginamos que la aplicación es una montaña (ver Figura 1). El usuario tiene dos posibilidades diferentes de comportarse respecto a la montaña. Un primer estado es el inicial, un usuario pretende escalar de 1 en 1 los niveles de nuestra montaña, pasando por cada uno de ellos sin excepción.

Sin embargo, la aplicación permite la posibilidad a un usuario avanzado de colocarse en un punto más alto de la montaña, sin necesidad de escalar todos los niveles.

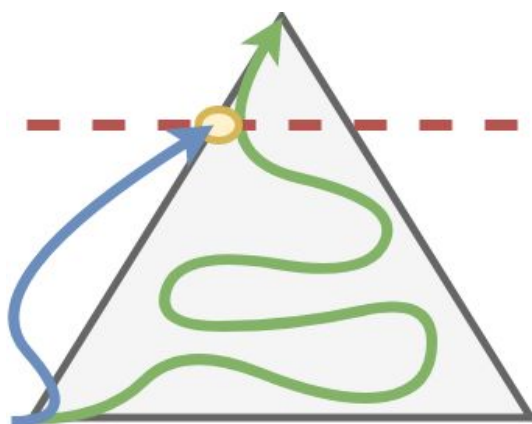


Figura 1. Formas de obtener nivel en la herramienta

Como ilustra el ejemplo anterior, el usuario de nuestra aplicación varía en torno a su nivel y en cuanto a su tipo de perfil. Puede ser un perfil profesional y querer mejorar su conocimiento en este ámbito en su entorno profesional, o puede ser un perfil de usuario normal, que únicamente quiere mejorar o aprender nuevos conocimientos. A su vez puede ser tanto un usuario con un conocimiento ya experimentado, como un usuario que empiece a interesarse por el tema de la autodefensa digital. Por esta razón se pensó en dividir los artículos de nuestra herramienta en torno a niveles. Estos niveles definen una serie de privilegios en la aplicación, tales como poder contribuir mediante la creación de nuevos artículos o la modificación de los ya existentes. Para que un usuario pueda conseguir aumentar su nivel, se pensó la incorporación de las pruebas de nivel. Un usuario debe demostrar sus conocimientos mediante la resolución de estas pruebas, cuyo contenido son una serie de preguntas basadas en el tipo de perfil del usuario y su nivel.

Se incorporó la posibilidad de que un usuario experimentado se registrase en nuestra aplicación. Con ello surgió la problemática de que este usuario comenzara con el nivel más bajo (uno). Por ello se implementó una primera prueba de nivel para aquel usuario que por primera vez se registre en la aplicación. Deberá acreditar su nivel respondiendo correctamente a las preguntas para conseguir avanzar de nivel, hasta un nivel máximo de 5 puntos. La Figura 1 muestra las diferentes maneras de aumentar nivel en la herramienta.

Otra de las ideas principales ha sido la de hacer que nuestra herramienta fuese colaborativa. Esto se ha conseguido mediante la posibilidad de generar nuevas propuestas de artículos o proponer la modificación de los ya existentes. Estas **contribuciones** por parte de los usuarios serán valoradas por el resto, y en el caso de obtener una valoración positiva por la mayoría de los usuarios, pasará a formar parte de los artículos consolidados en la aplicación. El control máximo sobre el contenido será gestionado por los administradores, quienes tendrán la última palabra sobre todo el contenido de la aplicación.

Otra manera de colaborar es la contribución a través de GitHub, ya existe un repositorio disponible para que cualquier usuario de esta aplicación contribuya en la funcionalidad de la herramienta **KapiSolution**.

KapiSolution

El nombre de la aplicación surge tras la búsqueda de un animal para el logo que representara la autodefensa. Qué mejor animal que un erizo. Kapi significa **Erizo** en la lengua Maorí, buscamos un idioma no muy común y nos gustó Kapi. **KapiSolution**.

1.3. Objetivos

Todo lo comentado en la subsección anterior nos lleva a definir los siguientes objetivos de la herramienta desarrollar en este Trabajo Fin de Grado, **KapiSolution** :

1.3.1. Formación acerca de la autodefensa digital

Es el objetivo principal. Reunir en un mismo sitio web información acerca de la autodefensa digital para distintos tipos de usuarios. **KapiSolution** tiene como objetivo formar a **diferentes perfiles profesionales** acerca de la autodefensa en su interacción diaria con las herramientas digitales. Para ello, la información se ha estructurado en una serie de roles y niveles. Porque no es lo mismo para un abogado protegerse en sus comunicaciones diarias cifrando el correo, que para un periodista montar una VPN estando trabajando en un país que no permite indexar las búsquedas a distintos sitios web. Además, no todos los roles (o perfiles que se pueden acercar al mundo de la autodefensa digital) tienen el mismo background en tecnología. [\[RF01\]](#)

También ha sido nuestro objetivo diferenciar que un abogado no tiene por qué tener conocimientos en informática, ni un periodista a su vez. Por ello, se presentan **diferentes niveles** junto con sus respectivas pruebas de nivel para disminuir la agresividad en una primera interacción con la herramienta. Esto forma parte de la importancia en **estructurar el contenido** para que la **búsqueda** de los recursos sea lo más **sencilla** posible. [\[RF05\]](#)

Además, para fijar los conocimientos adquiridos se pueden realizar dos tipos de pruebas de nivel:

- En primer lugar, cada vez que un usuario nuevo se registre, realizará una primera prueba para determinar su nivel inicial.
- Por otro lado, la herramienta ofrece la posibilidad de mejorar de nivel en la aplicación mediante pruebas específicas para cada nivel. En el apartado de pruebas de nivel, el usuario podrá acceder a la prueba del nivel siguiente al suyo actual. Si el resultado de la prueba es correcto, se aumentará un nivel al usuario, así hasta completar las 9 pruebas de evaluación existentes en el sistema. [\[RF07\]](#)

Cada prueba consta de una serie de preguntas específicas para cada rol y nivel.

1.3.2. Herramienta colaborativa

Otro de los objetivos que ha dirigido el desarrollo de este trabajo en todo momento ha sido hacer que la aplicación pueda mejorarse (programación y contenidos) de forma colaborativa. Esta colaboración está jerarquizada, es decir, un usuario que ha conseguido una serie de privilegios puede colaborar activamente en distintas fases de la herramienta. Estas fases comprenden desde colaborar programáticamente en el proyecto mediante la colaboración en el

repositorio de GitHub [\[RNF03\]](#), como realizar nuevas propuestas de artículos, o modificar los ya existentes. [\[RNF02\]](#)

Los privilegios se consiguen a través del nivel del usuario en la herramienta. Si es capaz de superar las pruebas de nivel, tendrá privilegios de un nivel superior. En ese momento, el usuario podrá sugerir modificaciones en los artículos de su mismo nivel al igual que proponer nuevos artículos (contribuciones). En ese momento, las modificaciones sugeridas por el usuario pasarán a ser evaluadas por el resto de usuarios, que decidirán con sus votos, si dichas contribuciones pasan a ser artículos consolidados.

La Figura 2 presenta distintas formas de colaborar:






FORMAS DE COLABORAR			
ADMINISTRADORES	USUARIOS REGISTRADOS	USUARIOS GITHUB	USUARIOS NO REGISTRADOS
 <ul style="list-style-type: none"> • CREAR, MODIFICAR Y ELIMINAR CONTENIDO • GESTIÓN DE USUARIOS • GESTIÓN DE LOS PERFILES DE LA HERRAMIENTA (INFORMÁTICO, ABOGADO, PERIODISTA) • GESTIÓN DE LAS PRUEBAS DE NIVEL 	 <ul style="list-style-type: none"> • CREAR, MODIFICAR ARTÍCULOS • VOTAR COLABORACIONES DE USUARIOS • REALIZAR PRUEBAS DE NIVEL 	 <ul style="list-style-type: none"> • PUEDE COLABORAR A TRAVÉS DEL REPOSITORIO DE GITHUB https://github.com/kapisolution/kapi_solution 	 

Figura 2. Formas de colaborar según tipo de usuario

Las formas de colaborar en la herramienta se pueden dividir en función de los tipos de usuarios que interaccionan con la misma. Existen dos maneras diferentes de colaborar en la aplicación.

- La primera forma de colaborar está enfocada al contenido de la aplicación. Todos aquellos usuarios que estén registrados en la aplicación podrán contribuir al crecimiento de la misma creando nuevos artículos o modificando los ya existentes. En ambos casos, se convertirán en **contribuciones** que el resto de usuarios registrados podrán votar. En caso de que la contribución sea aceptada por gran la mayoría de los usuarios, se incorporarán a la aplicación como artículos.
- La segunda manera de colaborar con la herramienta es a través de GitHub. Existe un repositorio en el siguiente enlace [\[E1\]](#) donde cualquier usuario que lo desee podrá proponer cambios a nivel programático en la aplicación. Estos cambios serán aceptados o desechados por los administradores de la organización **KapiSolution**.

Los **roles o tipos de usuarios** que se van a relacionar con la aplicación son:

- **Administradores:** Aquellos usuarios que tienen la función de gestionar usuarios, eliminando cuentas en el caso de un uso indebido. Tienen poder sobre las pruebas de

nivel, se encargan de realizar dichas pruebas e incorporarlas en el sistema. Se encargan del uso correcto de la herramienta.

- **Usuarios no registrados:** Son los usuarios que se relacionan con la herramienta de una forma únicamente de consulta. No tienen permisos de colaboración.
- **Usuarios registrados:** Hablamos de los usuarios que tienen habilitados ciertos lugares de la herramienta que un usuario sin registrar desconoce. Tiene acceso a crear contenido, modificarlo en base a sus permisos y acceso a las pruebas de nivel. Además de un perfil propio con sus datos y contribuciones en la aplicación.
- **Usuarios GitHub:** Se relacionan con la herramienta mediante mejoras a nivel programático. Interaccionan de una manera diferente con la aplicación, pero tienen relación con ella.

GitHub es el lugar donde se encuentra nuestro repositorio de **KapiSolution**. Para poder contribuir en GitHub en el proyecto, un usuario debe seguir una serie de pasos.

Cómo colaborar en un proyecto en GitHub

- Fork del repositorio
- Clonar el repositorio
- Actualizar la rama master
- Hacer los cambios
- Hacer un pull request

Fork del repositorio

El primer paso es hacer “Fork” del repositorio

Clonar el repositorio

Después de tener el repositorio en nuestra cuenta, seleccionar la dirección del repositorio SSH o HTTP y clonar:

```
git clone https://github.com/User/NombreRepo.git
```

Dentro de la carpeta que genera, comprobar la URL del repositorio:

```
git remote -v
```

Antes de realizar modificaciones agregar la URL del repositorio de origen del proyecto:

```
git remote add upstream https://github.com/User/RepoOriginal(Forkeado)
```

Actualizar la rama master

```
git pull -r upstream master
```

Crear una rama

git checkout -b feature-nombre-rama

Hacer cambios

Realizar todos los cambios que se desea hacer al proyecto.

Agregar los archivos y hacer un commit

Después de realizar el commit hacer el push hacia nuestro repositorio indicando la rama que hemos creado.

git push origin feature-nombre-rama

Hacer un Pull Request

Hacer click en “Compare & Pull Request”

Escribir cambios del Pull Request, y si todo es correcto enviar con el botón Send Pull Request.

Esperar a que el dueño del repositorio lo revise, acepte y mezcle en la rama correspondiente.

1.4. Plataformas similares

1.4.1. Wikipedia

Wikipedia [\[E2\]](#) se define como una enciclopedia libre, políglota y editada colaborativamente. Es administrada por la Fundación Wikimedia, una organización sin ánimo de lucro cuya financiación está basada en donaciones [\[10\]](#).

Cualquier usuario puede colaborar en wikipedia, existe un asistente para la creación de nuevos artículos donde se explica paso a paso cómo contribuir en la herramienta. El asistente distingue entre tipos de usuarios (usuarios novatos, representantes de una organización, usuario experimentado...) y, para cada tipo de usuario tiene una sección dedicada a la creación de nuevo contenido.

Existen los usuarios registrados, actualmente más de 5 millones, de los cuales menos del 0,4% son activos. [\[11\]](#)

Es similar a nuestra herramienta ya que comprende gran cantidad de información y está administrada por personas con alto nivel de conocimiento. Estos usuarios se eligen mediante votación democrática, y representan no son más del 0,5% de los usuarios activos.

También existen bots que realizan algunas funciones tediosas para los administradores como pueden ser ajustes internos del código, corrección de faltas ortográficas...

1.4.2. Guía de autodefensa catalana

Guía de autodefensa catalana [3] se define a sí misma como:

“Esta guía tiene como objetivo explicaros cómo puede mejorar tu privacidad y seguridad en Internet. Si tiene algún comentario para mejorarla, nos lo puede hacer llegar por correo electrónico a info@softcatala.org. [12]

Si no se especifica lo contrario, todos los programas mencionados en esta guía se encuentran disponibles en catalán.”

Podríamos decir que es una plataforma con una idea similar a la nuestra. Reunir en un sitio web artículos relacionados con la autodefensa digital. Se podría decir que es colaborativa en cuanto un usuario puede sugerir al correo contenido diferente, pero no podemos decir que su objetivo sea colaborativo en sí.

Los artículos presentes en la aplicación tratan sobre la protección en la navegación web. Existe un artículo para navegar a través de Tor, un artículo dedicado a la mensajería segura a través de diferentes tipos de aplicaciones como WhatsApp, Telegram, signal...Una sección para el cifrado de datos personales, etc...

Los contenidos están centrados en el anonimato al navegar a través de internet, no pienso que tenga un ánimo educativo ni colaborativo en sí, por tanto dista de la idea de nuestra herramienta.

1.4.3. Security Education Companion

Security Education Companion [4] se define en su página web como un recurso web para que educadores se formen en diferentes contenidos relacionados con la autodefensa digital. [13]

Presenta una serie de temas diferenciados en niveles para la consulta por parte del usuario. Se relaciona con nuestra idea en presentar diferentes contenidos diferenciados por niveles, y por agrupar en un mismo sitio, conocimiento correspondiente a la autodefensa digital.

La diferencia fundamental respecto a nuestra aplicación es el carácter colaborativo, ya que en esta herramienta no se distinguen roles ni se especifica la posibilidad de generar contenido para la aplicación. No tiene una sección de usuarios registrados, no existen, permite todas sus funcionalidades a cualquier usuario que accede a su sitio web.

Tiene una función educativa ya que su contenido se divide en lecciones y se definen como un recurso web para educadores. Se diferencia con respecto a **KapiSolution** en que el usuario no puede realizar pruebas de nivel

1.4.4. Comparativa entre aplicaciones

Como se ha descrito anteriormente, existen varias aplicaciones con características comunes a la nuestra, pero ninguna reúne el mismo número de funcionalidades. En la siguiente tabla (Tabla 1) se puede observar una comparativa entre nuestra aplicación y el resto de las descritas en el punto 1.4 en base a las diferentes funcionalidades que presenta **KapiSolution**.

Se puede observar que Wikipedia es la que más se puede acercar a nuestra idea por la función colaborativa y sus roles, mientras que el resto sólo coinciden en una funcionalidad con respecto a nuestra herramienta.

Aplicación	Roles	Niveles	Pruebas de nivel	Colaborativa
Guia de autodefensa catalana				
Security Education Companion				
Wikipedia				
KapiSolution				

Tabla 1.Comparativa entre herramientas similares

1.4.5. Tecnologías similares

A continuación se exponen una serie de tecnologías que comparten una funcionalidad con nuestra aplicación.

- **WordPress:** es un sistema de gestión de contenidos enfocado a la creación de cualquier tipo de web. Originalmente alcanzó una gran popularidad en la creación de blogs, para convertirse con el tiempo en una de las principales herramientas para la creación de páginas web comerciales. Está desarrollado en el lenguaje PHP para entornos que ejecuten MySQL y Apache, bajo licencia GPL y es software libre. [14]. Como se puede observar, comparte tecnologías de desarrollo con nuestra aplicación y permite la creación de contenido, en este caso mediante la creación de blogs.
- **Moodle:** Moodle es una herramienta de gestión de aprendizaje (LMS), o más concretamente de Learning Content Management (LCMS), de distribución libre, escrita en PHP. Está concebida para ayudar a los educadores a crear comunidades de aprendizaje en línea [15]. Comparte con nuestra aplicación el uso de PHP como lenguaje y la posibilidad de otorgar al usuario un sitio de aprendizaje en diferentes ámbitos.
- **WikiMedia:** Como se describe en su página web de Wikipedia, “la misión de la Fundación Wikimedia es facultar y animar a la gente de todo el mundo a reunir y desarrollar contenido educativo neutral bajo una licencia de contenido libre o en el dominio público, y a difundirla de manera efectiva y global. En colaboración con una red de capítulos, la Fundación proporciona la infraestructura esencial y la estructura organizativa para el apoyo y desarrollo de proyectos [16]. Comparte con nuestra herramienta el deseo de reunir conocimiento y difundirlo. Proporcionando una infraestructura para la creación y consolidación del contenido.

2. Metodología y plan de trabajo

2.1. Metodología de trabajo

El proyecto se ha desarrollado por dos estudiantes Se han diferenciando claramente las tareas a realizar por cada integrante del proyecto con el fin de ser lo más autónomo posible. Además, esta división clara de las tareas a realizar permite la presentación de dos trabajos fin de grado independientes en diferentes convocatorias.

En cuanto a las tareas que he asumido durante el desarrollo del proyecto, comentar el diseño del modelo de datos de la aplicación, el desarrollo completo de la funcionalidad de la herramienta y el desarrollo frontal de la web. Mi compañero me ha dado apoyo en la interacción con la base de datos mediante consultas y cargas masivas de datos para pruebas.

A continuación se detallan las tareas que he realizado durante el desarrollo del proyecto:

- Desarrollo de la **parte frontal de la web**.
- Desarrollo de la **funcionalidad** de la herramienta:
 - Desarrollo de las funcionalidades relacionadas con el usuario:
 - Registro.
 - Perfil de usuario.
 - Edición de perfil de usuario.
 - Pruebas de nivel.
 - Desarrollo de las funcionalidades relacionadas con la comunidad:
 - Creaciones de nuevos artículos
 - Modificaciones de artículos.
 - Consolidación de contribuciones.
 - Votaciones de las contribuciones.
- Modelo de base de datos.
- **Validaciones** por nivel y por perfil profesional.
- Desarrollo de las **restricciones** de contenido a usuarios.
- Funciones **JavaScript** de la aplicación:
 - Ordenación, muestra de resultados, validaciones, integración con el editor de texto **TinyMCE**.

Para el control de versiones hemos utilizado **Git**. Entraremos algo más en detalle en relación a Git y GitHub [\[5\]](#) más adelante. Git es una herramienta que permite mantener diferentes versiones del proyecto ubicadas en distintas ramas y posteriormente mezclarlas en una rama master.

La metodología de trabajo seguida en relación al control de versiones ha sido mantener 4 ramas diferenciadas (master, rama mezcladora, rama integrante 1, rama integrante 2). En la figura 3 se pueden distinguir estas 4 ramas. El por qué de usar estas cuatro ramas es debido a lo siguiente:

Existe una rama para cada integrante del proyecto. El desarrollo en cada rama es independiente, y el único requisito es no tocar los mismos ficheros por parte de los programadores.

Cuando algún programador finaliza su desarrollo, incorpora en la rama mezcladora sus cambios, estando disponibles para los otros programadores. Este último, cuando termina su desarrollo, actualiza la rama mezcladora incorporando sus cambios. En este momento, development (rama mezcladora) ya está lista para incorporar a master (rama funcional) el desarrollo de ambos integrantes.

Es importante destacar que ningún usuario que no pertenezca a la organización **KapiSolution** [1] no podrá realizar ningún cambio en la rama master ni en el resto de ramas del proyecto sino que deberá colaborar realizando un fork del proyecto. (Ver sección cómo colaborar en un proyecto GitHub en el punto [1.3.2](#)).

A continuación se explican las ramas ya mencionadas.

- **master**: Rama funcional, se encuentra la última versión funcional del proyecto.
- **development**: Rama donde se incorporan los cambios de cada una de las ramas de los integrantes, cuando cada uno de ellos finaliza su desarrollo.
- **rama 1, rama 2, ... rama i**: Ramas dedicadas cada desarrollo individual de las tareas por parte de cada integrante del proyecto.

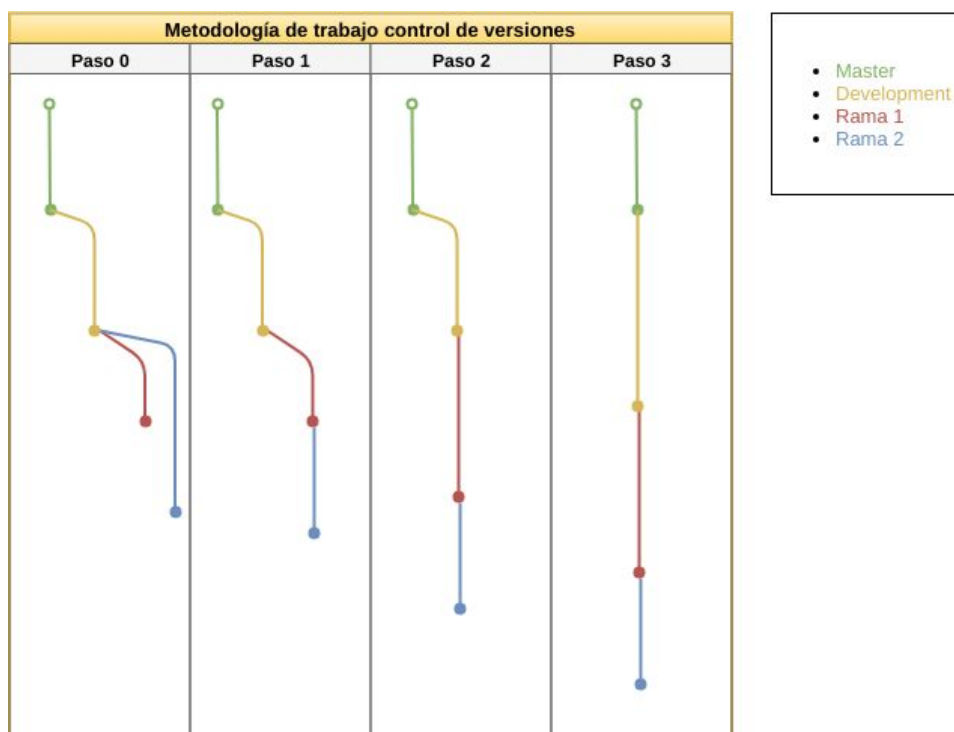


Figura 3. Metodología de trabajo para el control de versiones

En nuestro repositorio de GitHub se encuentra, en el siguiente enlace [\[9\]](#), todas las tareas desarrolladas durante el proyecto.

De igual manera, GitHub te permite realizar un seguimiento del proyecto teniendo una visión global del mismo a través del seguimiento visual de las ramas, fechas de commits y la posibilidad de ver el trabajo realizado en cada fecha, una wiki explicativa del proyecto y la posibilidad de realizar una GitPage [\[6\]](#), un host que te ofrece **github** con una serie de plantillas para dar una visión global de la herramienta y que se ha incorporado en este proyecto.

Como la herramienta está pensada para permitir la colaboración no sólo en la parte de edición de las páginas y test si no también en la parte de desarrollo de la arquitectura de la herramienta, para fomentar la colaboración conviene disponer al menos de una wiki actualizada donde se explique cómo colaborar en la parte de desarrollo.

2.2. Arquitectura

El proyecto se ha realizado bajo el estándar **LAMP (Linux, Apache, MySQL, PHP)**. Es un estándar que comprende el uso de diferentes herramientas de programación para el desarrollo de aplicaciones web.

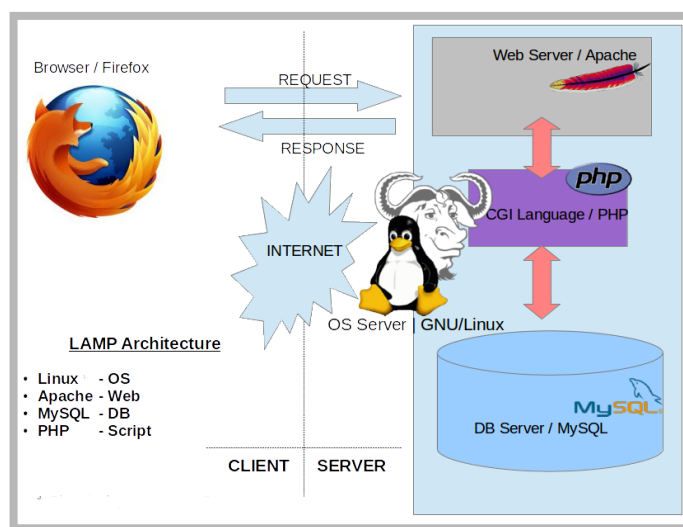


Figura 4.Arquitectura de una aplicación bajo el stack LAMP

Linux (núcleo del sistema operativo libre tipo Unix) otorga un entorno favorable para el tipo de aplicación que se ha desarrollado. En nuestro caso, nos ha permitido manejar y configurar el servidor **Apache** de una manera sencilla y accesible. Además de lanzar el sistema gestor de base de datos **MySQL** sin mayor complicación. Nos ha permitido tener una visión total del proyecto, desde la configuración del servidor hasta el resultado final de la aplicación. Hemos

buscado controlar los máximos aspectos de la aplicación, además de familiarizarnos con este modo de desarrollo. [\[17\]](#)

Como editor de código he desarrollado utilizando **Visual Studio Code**, un editor de código libre, como punto a favor a destacar ha sido su integración con **Git**, permitiendo controlar las versiones de los archivos modificados desde el propio editor.

En cuanto a la arquitectura de la propia aplicación, hemos distinguido diferentes bloques.

- **Backend:** Es la parte alrededor de las consultas a la base de datos, en algunos casos se seleccionan y envían los datos para mostrar a través del navegador, y en otros casos se encarga de las inserciones y actualizaciones de tuplas en base de datos. Se han desarrollado mecanismos de validaciones de datos previos a su inserción en la base de datos.
- **CSS:** Hojas de estilos. Se encuentran los archivos encargados de dotar de estilo a la aplicación.
- **DOC:** Ficheros de base de datos, se almacenan los ficheros JSON con los exámenes. Además se guardan los ficheros correspondientes a las imágenes, tanto de usuario como de artículos y contribuciones. También se encuentran los script .sql de la base de datos.
- **Frontend:** En esta sección se encuentran los archivos donde se desarrollan las distintas vistas de la herramienta. Se incluye código PHP para las restricciones de los usuarios a determinadas zonas de la aplicación y para el manejo y manipulación de los datos que se muestran en la web.
- **JS:** Funciones JavaScript que otorgan de funcionalidad extra a la aplicación. Se han desarrollado funciones de ordenación, de filtrado de resultados y validaciones como puede ser el registro en la parte frontal de la aplicación.

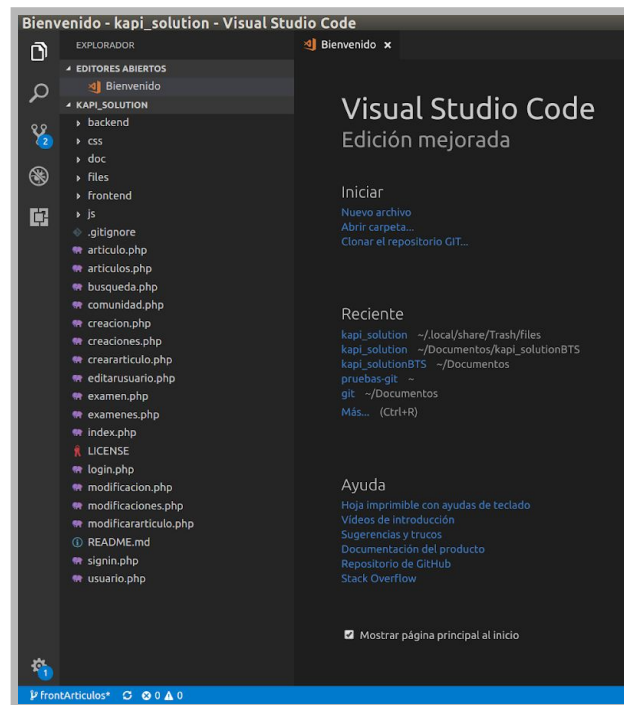


Figura 5. Arquitectura de trabajo durante el desarrollo de la aplicación

La herramienta está desarrollada bajo un sistema de base de datos relacional cuya arquitectura se detalla a continuación. Es un sistema pensado para el **mantenimiento mínimo por parte del administrador**. Esto es así ya que la aplicación es nutrida de **contenido** por parte de los usuarios mediante las **contribuciones**. Se ha desarrollado el modelo de datos para que en el momento que una contribución sea aceptada por parte de los usuarios mediante el sistema de **votaciones**, esta se incorpore inmediatamente a los artículos **consolidados** en la web. De esta manera, el administrador no tiene que realizar la inserción de nuevo contenido en la aplicación, sino que son los propios usuarios, mediante el sistema de contribuciones los que otorgan un **crecimiento** a la aplicación. Esto se consigue gracias al desarrollo de ficheros php que se encargan de, en el momento que se realiza una votación de una contribución, comprobar si cumple el criterio de consolidación de contenido y, en ese caso, de la posterior actualización de un artículo si la contribución es una modificación, o bien, de la creación de un nuevo artículo en base de datos si la contribución es una nueva creación. Se explica en el punto [4.2.3](#) Votaciones sobre contribuciones de los usuarios.

Esto se consigue también mediante el diseño de un modelo de datos que permita almacenar contribuciones y artículos en entidades separadas pero con una estructura común.

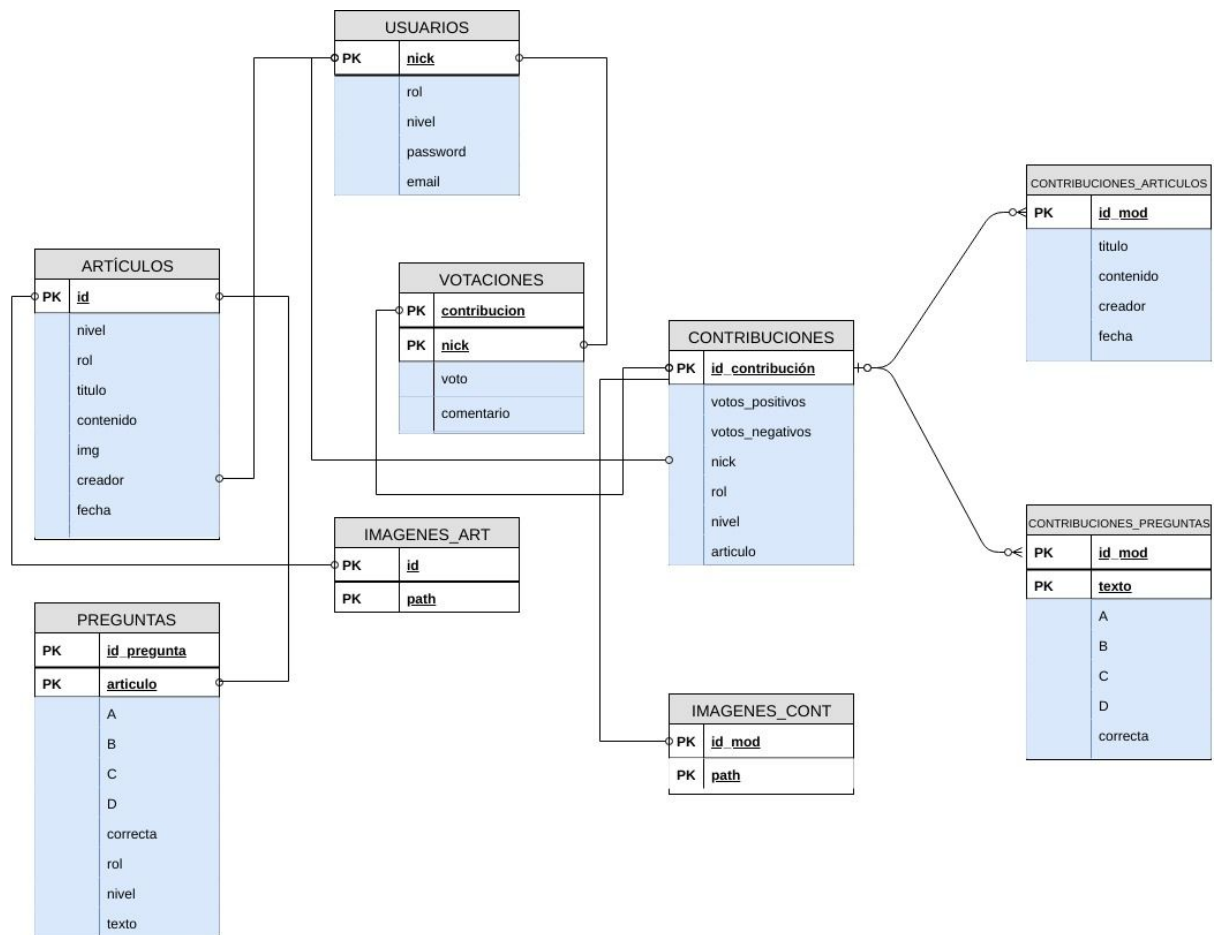


Figura 6. Estructura de la base de datos

Como se puede ver en la Figura 6, distinguimos diferentes entidades en la base de datos, aquellas necesarias para representar la lógica de la aplicación. Los usuarios podrán ver los artículos, votar y realizar contribuciones. Los artículos se relacionan con sus preguntas (cada artículo tiene N preguntas) y sus imágenes (cada artículo tiene N imágenes) mediante el id de artículo. Las contribuciones también tienen su relación con una tabla de imágenes, siguiendo el mismo criterio que para los artículos. Estas tablas referentes a las imágenes únicamente almacenan una ruta del servidor donde se encuentran realmente las imágenes. Este mecanismo se ha desarrollado para evitar una sobrecarga en la base de datos. Además, las contribuciones se han diferenciado en dos tablas distintas (artículos y preguntas) siguiendo el modelo de herencia en bases de datos con respecto a la tabla modificaciones. Ambas tablas comparten los atributos referentes a los votos, el nick del creador, nivel, rol y artículo, mientras que cada una de ellas tiene sus propios atributos.

2.2.1. Control de versiones (Git y GitHub)

Se ha mencionado ya el uso de esta herramienta con respecto a nuestro desarrollo. A continuación se comentan brevemente los motivos que respaldan esta decisión.

En primer lugar, Git facilita el control de versiones durante el desarrollo de un proyecto. Esto es así gracias a que registra cada cambio realizado durante el desarrollo del proyecto. De modo que permite recuperar cualquier versión en cualquier fecha sin importar en qué punto del desarrollo nos encontremos. Podemos aprovecharnos de esta utilidad tanto en el desarrollo de este proyecto (se realiza de forma independiente por dos estudiantes), como en un futuro cuando pueda haber más colaboradores trabajando en mejorar las funcionalidades de la herramienta.

Git es un sistema de control de versiones distribuido. Esta característica significa que los clientes no solo descargan la última copia instantánea de los archivos, sino que se replica completamente el repositorio, de esta manera, si un servidor deja de funcionar y existen sistemas que estaban colaborando a través de él, cualquiera de los repositorios disponibles en los clientes puede ser copiado al servidor con el fin de restaurarlo. Cada clon es realmente una copia completa de todos los datos. [\[18\]](#)

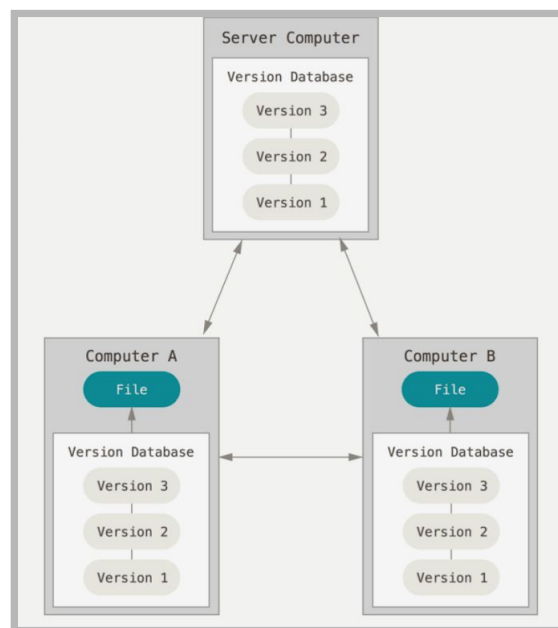


Figura 7. Sistema de control de versiones distribuido.

2.3. Especificación de requisitos

2.3.1. Requisitos funcionales

- **RF01 - Servir al usuario de nuevo conocimiento**

Se proporciona al usuario de artículos sobre autodefensa digital, por ello se pretende la mejora del conocimiento por parte del usuario en este ámbito.

- **RF02 - Restricción de contenido para usuarios no registrados**

La herramienta no permite el acceso a distintas zonas de contenido a aquellos usuarios que navegan sin credenciales en la aplicación.

- **RF03 - Restricciones de contenido para usuarios registrados**

Un usuario registrado podrá:

- Crear y/o editar contenido de un nivel igual o inferior a su rol
- Crear y/o modificar contenido de un rol igual al suyo
- Realizar exámenes de un nivel superior al suyo
- Votar contenido de su mismo rol y de un nivel igual o inferior al suyo

Un usuario registrado **no** podrá:

- Crear y/o editar contenido de un nivel superior al suyo
- Crear y/o editar contenido de un rol distinto al suyo
- Realizar exámenes de más de un nivel respecto a su nivel actual
- Votar contenido de un rol distinto al suyo y/o de un nivel superior al suyo

- **RF04 - Consolidación de nuevo contenido**

Al colaborar con nuevo contenido para la herramienta, el usuario debe saber que sus contribuciones pasan a ser valorados por el resto de usuarios, que son los que decidirán sobre la vida de dicha creación.

- **RF05 - Búsqueda de contenido**

El usuario puede realizar búsquedas a través de la navegación para el contenido correspondiente a artículos, usuarios, creaciones y modificaciones.

- **RF06 - Diferente interfaz de bienvenida para usuarios no logueados**

Se ofrecerá una visión global de la aplicación, así como la posibilidad de registrarse a aquellos usuarios que naveguen en ausencia de credenciales para la herramienta, mientras que para los usuarios logueados se redireccionará a la vista de artículos.

- **RF07 - Nivel de usuarios**

Todo usuario registrado tendrá asignado un nivel en la aplicación. Este nivel determinará el acceso a las pruebas de nivel, restricciones sobre las contribuciones (no podrá ni editar ni crear contenido superior a su nivel ni tampoco votar aquellas contribuciones que sean de nivel superior al suyo).

- **RF08 - Enriquecedor de texto**

Está incluida la herramienta **TinyMCE**, **WYSIWYG** (what you see is what you get) que permite otorgar al contenido de los artículos la posibilidad de dar formato al mismo.

- **RF09 - Subida múltiple de imágenes**

Para la tarea de generación de contenido para la web, se dota a la aplicación de la posibilidad de subir varias imágenes por artículo, de manera que dichas imágenes se almacenan en el servidor y no en la base de datos. Evitando así una sobrecarga del gestor de la base de datos.

- **RF10 - Zona de perfil de usuario**

El usuario tiene la posibilidad de manejar su información en una sección dedicada a el donde, además de editar sus datos, puede visualizar el contenido creado por él para la aplicación. Todos los usuarios tienen acceso al perfil de los demás usuarios, no así a la zona donde se editan los datos. Un usuario sólo podrá editar su propia información.

- **RF11 - Aplicación Responsive**

La herramienta está preparada para su uso en distintos dispositivos debido al desarrollo de la misma mediante el uso del framework Bootstrap que permite la visualización en distintos dispositivos como ordenador, móvil, tablet...

2.3.2. Requisitos no funcionales

- **RNF01 - Interfaz amigable**

La aplicación se ha desarrollado a través de herramientas actuales y acorde a una estética actual, de manera que un usuario sin un conocimiento especial está totalmente capacitado para usar la herramienta.

- **RNF02 - Posibilidad de colaborar**

La herramienta está desarrollada de manera que sea escalable acorde a los usuarios. Serán ellos los que controlen el contenido mediante las posibilidades que se les ofrecen y el sistema de votaciones que decidirá qué artículos pasan a ser consolidados como confiables.

- **RNF03 - Escalabilidad de la aplicación**

Existe el repositorio de GitHub **KapiSolution** [6] donde todo aquel que quiera dotar de un nuevo desarrollo a la web, lo puede hacer. La herramienta tiene una arquitectura diferenciada y preparada para nuevas líneas de desarrollo.

- **RNF04 - Usabilidad**

Se pretende que cualquier usuario sin un conocimiento especial sobre ningún ámbito pueda interactuar de una manera sencilla con la aplicación debido a su diferenciación en roles y niveles.

- **RNF05 - Portabilidad a otros entornos**

Se asegura el buen funcionamiento de la aplicación en distintos navegadores como Chrome, Mozilla Firefox, Internet explorer y Chromium.

2.4. Restricciones sobre perfiles profesionales y nivel de usuario

Este es un punto importante en la definición de la lógica de la aplicación. El sistema está pensado con una serie de restricciones que se definen a continuación.

- Un usuario **no registrado** no puede ver las modificaciones de artículos ni las creaciones de nuevos artículos.
- Un usuario **no registrado** no puede editar artículos.
- Un usuario **no registrado** no puede crear nuevo contenido.
- Un usuario **registrado** puede:
 - **Crear** contenido de **nivel inferior o igual** al suyo actual y **del mismo rol**.
 - **Modificar** artículos de **nivel inferior o igual** al suyo y **de su mismo rol**.
 - **Votar** contribuciones de su mismo **rol**.
 - **Realizar pruebas de un nivel por encima del suyo actual**.
- Un usuario **registrado no puede**:
 - **Crear** contenido de un **nivel superior** o de un **rol distinto**.
 - **Modificar** artículos de **nivel superior** al suyo o de **distinto rol**.
 - **Votar** contribuciones de **distinto rol**.
 - **Realizar pruebas de nivel inferior o superior en dos a su nivel actual**.

3. Tecnologías utilizadas

Como se ha comentado anteriormente, se ha desarrollado bajo el standard LAMP, donde las principales tecnologías son **Linux, Apache, MySQL y PHP**. A continuación se detallan las diferentes tecnologías utilizadas durante el proyecto:

- El **servidor HTTP Apache** es un servidor web libre y de código abierto, el más popular en cuanto a uso, sirviendo como plataforma de referencia para el diseño y evaluación de otros servidores web. En mi caso particular, he trabajado con una versión de apache (apache2), editando su respectivo archivo de configuración para la creación de host

virtuales con distintas IPs locales y pudiendo lanzar simultáneamente diferentes versiones del proyecto. [\[19\]](#)

- **MySQL** es un Sistema de Gestión de Bases de Datos (SGBD) relacional, que por lo tanto utiliza SQL, multihilo y multiusuario del que se estiman más de un millón de instalaciones. En nuestro caso se ha desarrollado un modelo de base de datos relacional, que se ha detallado en el punto de la arquitectura de la aplicación. [\[20\]](#)
- La aplicación está desarrollada mediante diferentes lenguajes de programación como son **PHP, HTML5, CSS y JavaScript**.

La mayor parte de la herramienta está desarrollada con **PHP**. Es un lenguaje de programación de propósito general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en un documento HTML en lugar, de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera el HTML resultante. [\[21\]](#) En nuestro caso, la aplicación de este lenguaje ha estado dedicada, entre otras tareas al **acceso y la consulta a la base de datos, al igual que para el manejo de datos en la parte frontal de la web. También se han implementado las restricciones de zonas de la aplicación a usuarios, mediante el uso de variables superglobales**. Estas variables, en nuestro caso las identificadas con `$_SESSION` se asocian al usuario y **se guardan en el navegador** durante el proceso de registro e inicio de sesión en la aplicación (ver código adjunto). Estas variables de sesión desaparecen al cerrar la sesión en el navegador, o bien cerrando el propio navegador.

```
$_SESSION["nick");//nick de usuario
$_SESSION["login");//identifica al usuario como logeado
$_SESSION["rol");//rol del usuario
$_SESSION["nivel");//nivel de usuario
$_SESSION["formularioNivel");//nos indica si ha realizado la prueba inicial
```

Para el control de acceso a usuarios no registrados a determinadas zonas de la aplicación, se usa la siguiente sentencia, que toma el valor true en caso de que el usuario esté identificado en la aplicación.

```
<?php
    if(!isset($_SESSION['login'])) {
        require 'accesoRestringido.php';
    }
?>

//$_SESSION[login] toma el valor true durante el proceso de registro,
//en otro caso su valor será false
```

HTML5 podemos decir que es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web,

como texto, imágenes, videos, juegos, entre otros. [22]. Nuestro uso del lenguaje ha estado dedicado a elaboración de la parte web de la herramienta, donde junto con el framework de desarrollo Bootstrap han dado forma a la parte frontal de la aplicación.

CSS es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado como es HTML en nuestro caso. Es muy usado para establecer el diseño visual de los documentos web e interfaces de usuario escritas en HTML o XHTML. [23]

Junto con **HTML y JavaScript**, **CSS** es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web, como es nuestro caso.

JavaScript es un lenguaje de desarrollo utilizado principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. Se ha incorporado en nuestro proyecto desarrollando diferentes funciones en este lenguaje para hacer nuestra herramienta más dinámica. [24]

En las siguientes figuras se muestran dos ejemplos del uso de JavaScript para el desarrollo de funciones. La primera figura muestra una función de ordenación de resultados y posterior pintado en el navegador, mientras que la segunda muestra la funcionalidad implementada para comprobar si es correcta una respuesta de una pregunta perteneciente a un artículo durante la visualización del mismo.

```
/*Funciones para la ordenacion de resultados*/
function comparatorFecha(a,b){
    if (a.fecha < b.fecha) return -1;
    if (a.fecha > b.fecha) return 1;
    return 0;
}
function comparatorNivelMen(a,b){
    if (a.nivel < b.nivel) return -1;
    if (a.nivel > b.nivel) return 1;
    return 0;
}
function comparatorNivelMay(a,b){
    if (a.nivel < b.nivel) return 1;
    if (a.nivel > b.nivel) return -1;
    return 0;
}
/*Llamadas a la ordenación y pintado de resultados*/
function ordenar(stringArray,criterio){
    if(criterio=="fecha"){
        $("#ordNivelMen").prop("checked", false);
        $("#ordNivelMay").prop("checked", false);
        stringArray=stringArray.sort(comparatorFecha);
    }else if(criterio=="nivelMen"){
```

```

    $("#ordFecha").prop("checked", false);
    $("#ordNivelMay").prop("checked", false);
    stringArray=stringArray.sort(comparatorNivelMen);
} else{
    $("#ordNivelMen").prop("checked", false);
    $("#ordFecha").prop("checked", false);
    stringArray=stringArray.sort(comparatorNivelMay);
}

for(var i=0; i<stringArray.length; i++){
    $("#rol"+i).html(stringArray[i].rol);
    $("#nivel"+i).html("Nivel "+stringArray[i].nivel);
    $("#titulo"+i).html(stringArray[i].titulo);

    $("#imagen"+i).attr("src","/files/img/usuario/"+stringArray[i].creador+".jpg");
    $("#creador"+i).html(stringArray[i].creador);
    $("#verArt"+i).attr("href","/articulo.php?id="+stringArray[i].id);

    $("#editarArt"+i).attr("href","/modificararticulo.php?id="+stringArray[i].id);
}
}

/*Funcion para comprobar respuesta de pregunta de artículo*/
function comprobarRespuesta(opcion,correcta,elemento){
    x = document.getElementsByClassName("label label-success respuesta");
    for (i = 0; i < x.length; i++) {
        x[i].className='label label-default';
    }
    y = document.getElementsByClassName("label label-danger respuesta");
    for (i = 0; i < y.length; i++) {
        y[i].className='label label-default';
    }
    if(opcion==correcta){
        $("#"+elemento).removeClass().addClass('label label-success respuesta');
    }else {
        $("#"+elemento).removeClass().addClass('label label-danger respuesta');
    }
}

```

3.1. TinyMCE

TinyMCE es un denominado **WYSIWYG**. Es una tecnología de **código abierto** usada para la incorporación de una herramienta de enriquecimiento de texto en una aplicación web. En nuestra herramienta se ha incorporado en el proceso de creación o modificación de un artículo. El usuario tendrá así la posibilidad de dar formato al contenido de la contribución. [\[RF08\]](#)

Para incorporar esta tecnología al proyecto, se ha incluido en la cabecera de los archivos donde se usa la siguiente librería JavaScript.

```
<script src="https://cloud.tinymce.com/stable/tinymce.min.js"></script>
```

Para activar el enriquecedor de texto, sólo hay que invocar la función **tiny.init()**, pasándole como argumento el selector (textarea en nuestro caso) dónde se va a iniciar. En la Figura 8 se puede ver un ejemplo de su uso en el bloque contenido de artículo durante el proceso de creación de una nueva contribución.

```
function divClickedContenido(area) {
  ...
  tiny.init({ selector: 'textarea#textoContenido' });
  ...
}
```

La documentación de dicho WYSIWYG está disponible en el enlace [\[7\]](#).

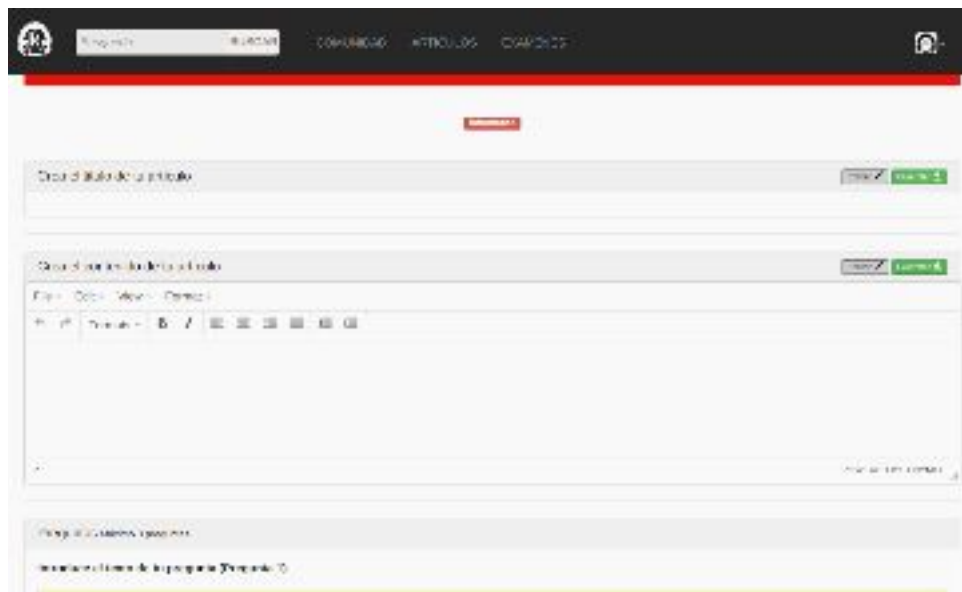


Figura 8. Ejemplo de uso de TinyMCE

3.2. Bootstrap

Cabe destacar la incorporación del framework de desarrollo de aplicaciones web **Bootstrap**.

Bootstrap es una librería de **código abierto** para diseñar aplicaciones web. Contiene plantillas HTML y CSS, formularios, botones, labels...componentes necesarios para el desarrollo frontal de cualquier aplicación. La principal característica es que maneja la visualización del contenido

en diferentes dispositivos (responsive) [\[RF11\]](#) abstrayendo al desarrollador de gran parte de este proceso. [\[25\]](#)

Se incorpora mediante librerías CSS y JavaScript.

```
<script  
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>  
<link rel="stylesheet"  
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
```

Este framework tiene desarrollada, entre otras características, la posibilidad de determinar en una sola línea, como se van a ver los componentes de la aplicación en distintos tamaños de pantalla. Para ello, Bootstrap incorpora el sistema **Grid**, dónde el programador define qué tamaño tendrán los componentes de la aplicación en distintos dispositivos.

Mediante este sistema, somos capaces de dividir la pantalla en filas de doce columnas, eligiendo cuántas columnas ocupa el componente en función de las dimensiones del dispositivo, las dimensiones admitidas son: xs (extra-small), sm (small), md (medium), lg (large).

A continuación se expone un ejemplo de este sistema incorporado en nuestra aplicación, donde se definen los distintos tamaños que adoptarán los bloques de la vista de artículos para distintos dispositivos. En este caso, para dispositivos grandes, se divide la pantalla en una fila de cuatro columnas para el bloque de filtrado de artículos y ocho para cada caja que contiene la información de cada artículo. Mientras que para dispositivos más pequeños, los mismos bloques descritos anteriormente ocupan ahora doce columnas cada uno, consiguiendo así una correcta visualización en cualquier dispositivo.

```
<div class="col-xs-12 col-sm-12 col-md-4 col-lg-4">  
  <div class="thumbnail">  
    <center><h4>Filtrar</h4></center><br>  
    ...  
  </div>  
  <div class="col-xs-12 col-sm-12 col-md-8 col-lg-8">  
    <div id="album" class="album py-5 bg-light">  
      <?php  
        for ($i = 0; $i < sizeof($articulos) ; $i++){  
          ?>  
          <div class="panel-group">  
            ...  
          </div>  
        }  
      </?php>  
    </div>  
  </div>
```

En nuestro caso, el uso de Bootstrap, aparte de servirnos de diferentes clases para la incorporación de componentes ya desarrollados por el framework, nos ha facilitado el desarrollo

responsive, permitiendo la visualización correcta de nuestra herramienta desde cualquier dispositivo, ya sea ordenador, móvil o tablet. Esto es una característica fundamental de nuestra aplicación, ya que la hace disponible en cualquier dispositivo, orientando la herramienta a poder utilizarse en diferentes ámbitos cumpliendo así el requisito funcional [\[RNF01\]](#).

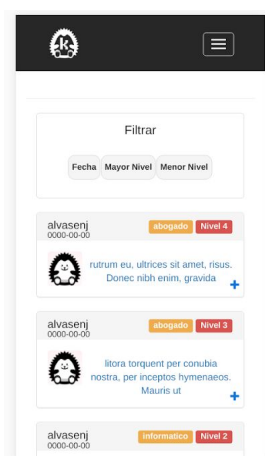
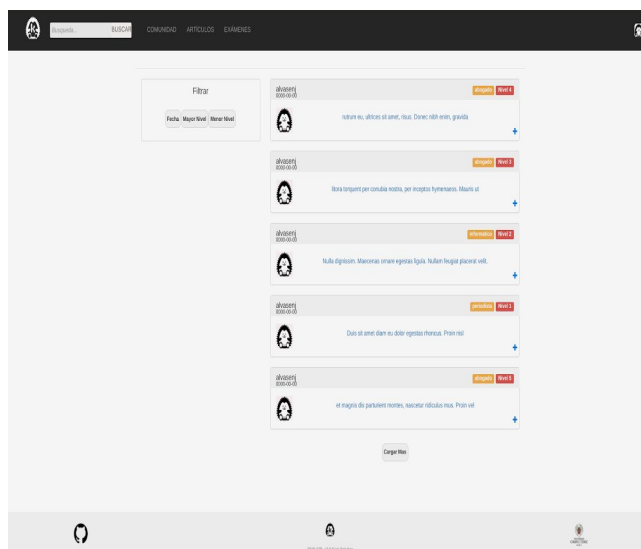


Figura 9. Comparativa vista de móvil con ordenador

Se ha desarrollado bajo la versión 3.3.7 de dicho framework. La documentación está disponible en [\[8\]](#)

4. Funcionalidad de la herramienta

En este apartado se pretende describir cuáles son las distintas funcionalidades de la herramienta **KapiSolution**. Se distinguen dos flujos de actividades. Por un lado las actividades relacionadas con los usuarios no registrados, como se muestra en la figura 16. Y por otro lado, las actividades relacionadas con aquellos usuarios que se han registrado previamente en la aplicación (Figura 17) . En las figuras se muestran también los archivos implicados en dichas actividades y las tablas de la base de datos que participan durante el flujo de la actividad.

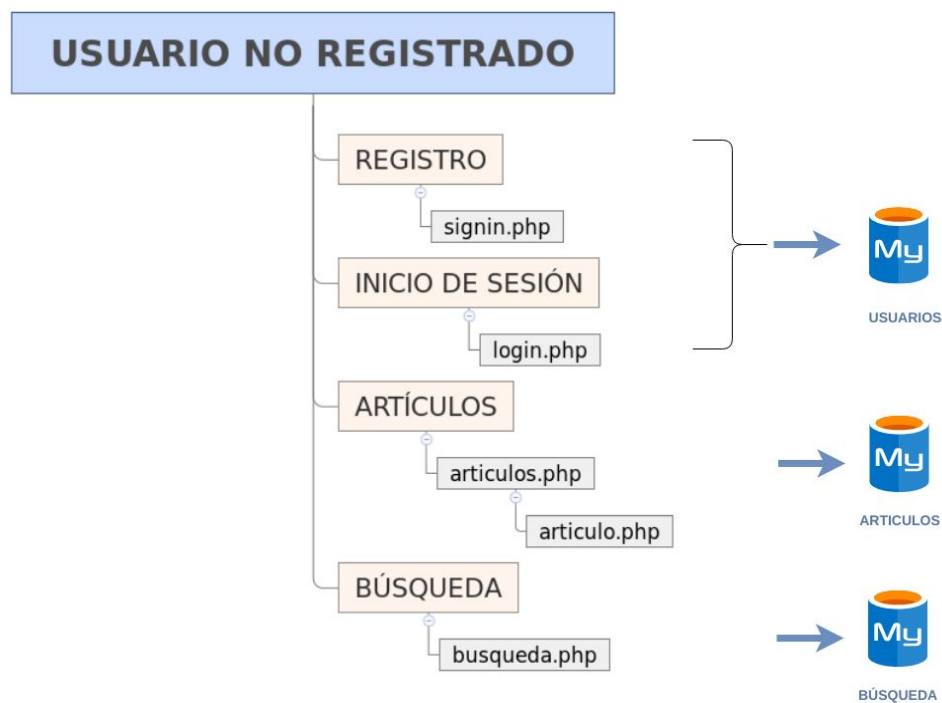


Figura 10. Flujo de actividades de la aplicación para usuarios no registrados

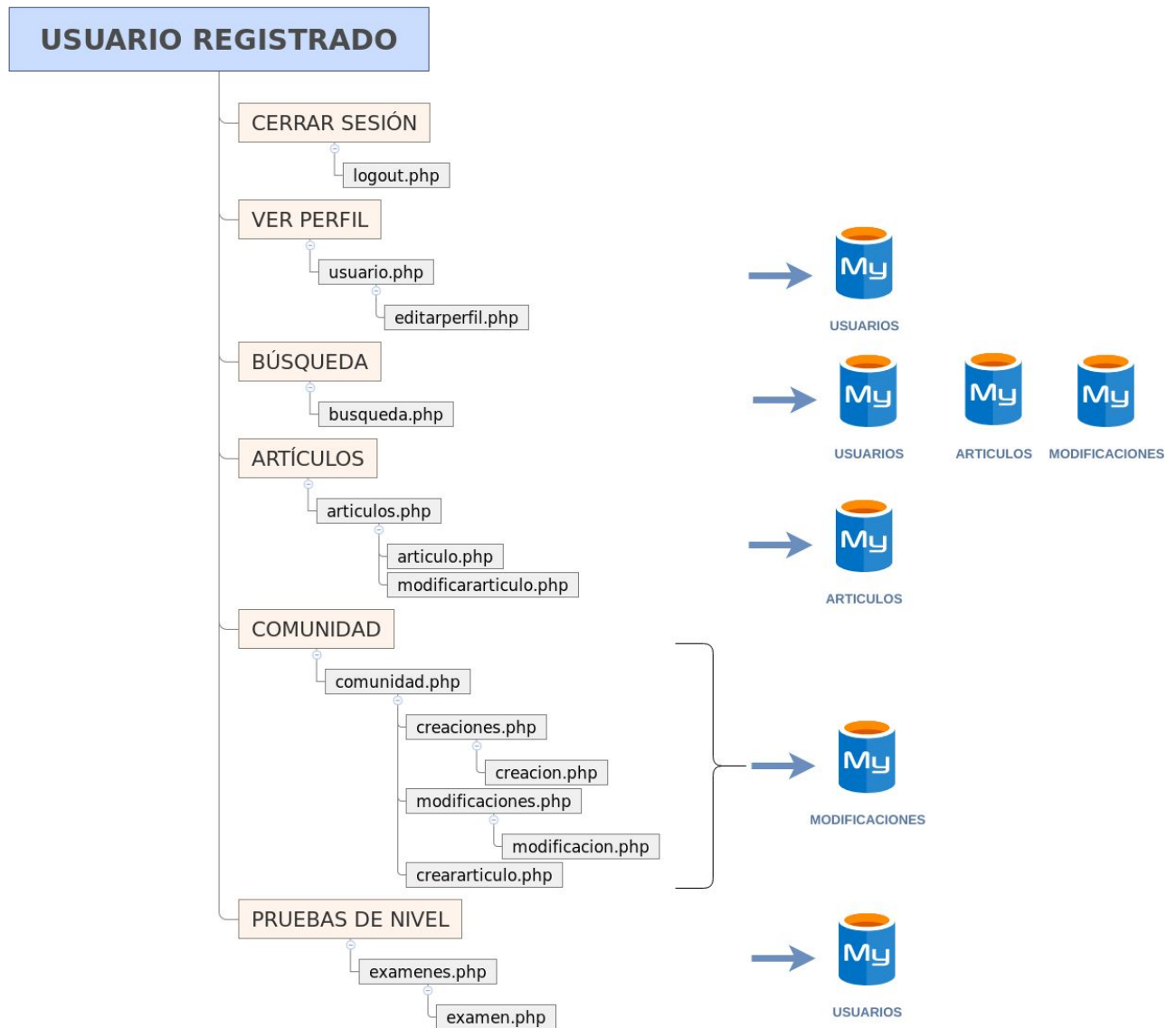


Figura 11. Flujo de actividades de la aplicación para usuarios registrados

Distinguimos por tanto:

4.1. Usuarios no logueados

Como se ha comentado, los usuarios que no se hayan registrado en la herramienta no van a tener a su disposición parte del contenido de la aplicación ni la funcionalidad completa. Un usuario no identificado puede navegar por la aplicación, comprobando el contenido de la misma y con la posibilidad de consultar los artículos, en cambio, no podrá consultar ni las creaciones ni las sugerencias de modificaciones de los artículos por parte del usuario. Tampoco podrá generar nuevo contenido ni realizar ninguna prueba de nivel. [\[RF02\]](#)

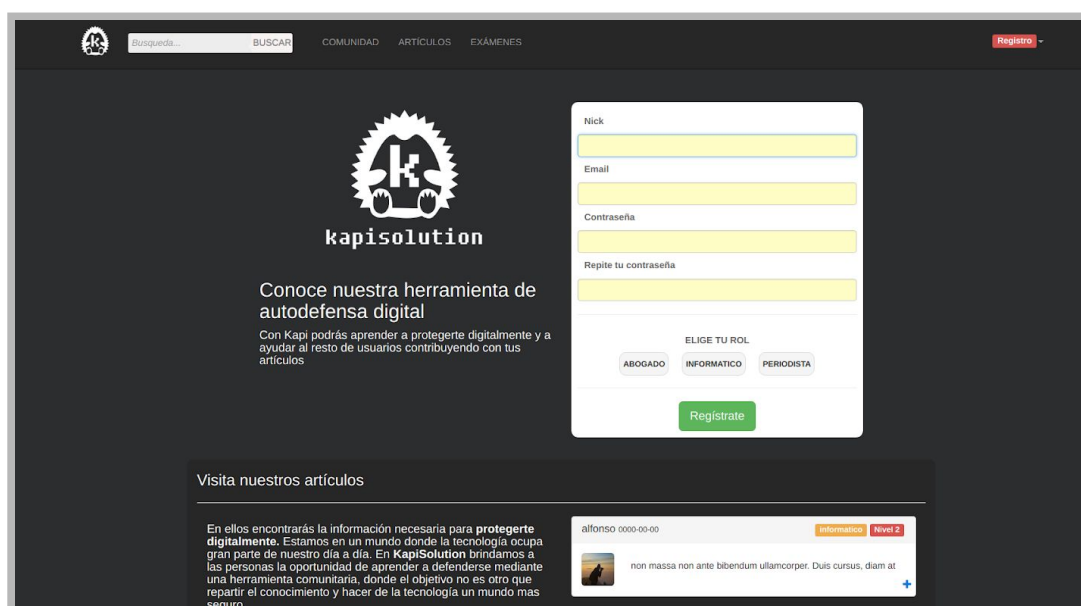


Figura 12. Aspecto de la herramienta para un usuario no logueado.

En primer lugar, este tipo de usuario se encontrará con una navegación capada. Esto es posible mediante las variables de sesión alojadas en el navegador, que permiten a la aplicación distinguir entre un usuario logueado y no logueado. Los usuarios que naveguen por la aplicación no verán parte del contenido ni tendrán la funcionalidad completa de la aplicación hasta el momento en el que estén identificados como usuarios de **KapiSolution**. La implementación de esta restricción está explicada en el apartado de tecnologías utilizadas, donde se menciona el uso de variables superglobales en php.

Las funcionalidades dedicadas a este tipo de usuario son las siguientes:

4.1.1. Registro

El registro se implementa mediante un formulario que recoge los datos mediante las variables de tipo `$_POST[]` de php. Posteriormente se realizan una serie de comprobaciones en la parte backend de la aplicación.



- Se comprueba si existe actualmente en la base de datos un usuario o un email como el introducido por el usuario. En ese caso se redirigirá a la página de registro con una variable de tipo \$_GET que se recoge para mostrar el mensaje de excepción al usuario.
- Se comprueba si las contraseñas introducidas son iguales. En ese caso se mostrará un mensaje de excepción como el descrito anteriormente a través de la variable \$_GET.
- Si todo ha sido correcto:
 - se codifica bajo cifrado md5.
 - se inserta el usuario en la base de datos.
 - se inicializan algunas variables de sesión.
 - se redirige a la página principal.

```
<!--
    Ejemplo uso $_SESSION[] en signin
-->
<?php
    session_start();
    require 'conexion.php'

    $sqlInsert = "INSERT INTO Usuarios(nick,rol,password,email)
VALUES('$nombre','$rol','$passdb','$email')";
    if (mysqli_query($con, $sqlInsert)) {
        $_SESSION["nick"] = $nombre;
        $_SESSION["login"] = true;
        header('Location: /');
    } else {
        echo("Error en la consulta");
    }
}
```

Posterior al proceso de registro y tras redirigir al usuario a la página principal, este debe realizar una primera prueba inicial para determinar su nivel en la aplicación. En la página principal se usa la variable de sesión **\$_SESSION[formularioNivel]** para mostrar la prueba inicial en caso de que no la haya hecho. Como esta variable no se inicializa en el registro, la primera vez que entre a la aplicación deberá completar la prueba. En el momento que el usuario realice dicha prueba, se calculará su nuevo nivel y se inicializará la variable de sesión **formularioNivel** de manera que cuando el usuario se dirija de nuevo a la página principal la prueba inicial habrá desaparecido. En el caso de que el usuario navegue por la aplicación sin realizar dicha prueba, esta opción estará disponible en la barra de navegación hasta que dicho usuario no la complete. También se le requerirá completarla al acceder a la página principal de la aplicación.

```
if($loggedIn) {
    if(trim($_SESSION['formularioNivel']) !=0) {
        header('Location:/articulos.php');
    }
    else{
        ?>
```

```

</script>
$('body').removeClass().addClass('modal-open');
$('body').attr("style","padding-right: 15px;");
$('nav').hide();
$('#footerBS.container-fluid.text-center.footerBS').hide();
</script>
<form id="testNivel" method="POST" action="backend/testNivel.php">
    <div class="modal fade in" role="dialog" id="myModal" tabindex="-1"
aria-labelledby="myModalLabel" style="display: block;">
    <div class="modal-dialog" role="document">
        ...
        <h4 class="modal-title">Responde a 5 preguntas para determinar tu
nivel</h4>
    </div>
    <div class="modal-body"><?php
        foreach($preguntas as $key => $pregunta) {
            <div class="panel panel-default">
                ...
            </div>
        <div class="modal-footer">
            <button type="submit" class="btn btn-default" >Enviar</button>
        </div>
        ...
    </div>
</form>
<?php
}
}

//backend/testNivel.php
<?php
    session_start();
    require 'conexion.php';
    $jsonText = file_get_contents("../doc/preguntasTestNivel.json");
    $preguntas = json_decode($jsonText, true);
    $puntos=0;
    //Comprobamos resultados
    foreach($_POST as $key => $value){
        if($preguntas[$key]['correcta'] == $value){
            $puntos+=$preguntas[$key]['puntos'];
        }else{
            $puntos-=$preguntas[$key]['puntos'];
        }
    }
    if($puntos < 0.5){
        $puntos = 1;
    }
    else if($puntos > 1 && $puntos < 4){
        $puntos = 2;
    }
}

```

```
}  
else if($puntos > 4 && $puntos < 6){  
    $puntos = 3;  
}else{  
    $puntos = 4;  
}  
$nombre=$_SESSION['nick'];  
//Actualizamos el nivel del usuario  
$sqlInsert = "UPDATE Usuarios SET nivel = '$puntos' WHERE nick = '$nombre'";  
if (mysqli_query($con, $sqlInsert)) {  
    $sql = 'SELECT * FROM Usuarios WHERE nick="'.$nombre.'";'  
    $resultado = mysqli_query($con, $sql) or die("Error por nick en consulta  
sobre la tabla Usuarios");  
    $usuario = array();  
    //Inicializamos variable de sesion prueba inicial  
    $_SESSION["formularioNivel"] = true;  
    //Actualizamos variable de sesión nivel  
    $_SESSION["nivel"] = $puntos;  
    header('Location: /');  
} else {  
    echo("Error en la consulta");  
}  
include 'desconexion.php';  
?>
```

La funcionalidad de la prueba inicial es la misma que la descrita en el apartado de [pruebas de nivel](#).

4.1.2. Inicio de sesión

El inicio de sesión, al igual que el registro, se implementa mediante un formulario que recoge los datos mediante las variables **\$_POST[]** de php. Posteriormente se realizan una serie de comprobaciones en la parte backend de la aplicación.

- Se comprueba si existe en base de datos el usuario introducido.
- Se comprueba si la contraseña introducida corresponde con la contraseña en base de datos del usuario.
- Si todo ha sido correcto:
 - Se inicializan variables de sesión para el control de acceso y permisos a usuarios.
 - Se redirige a la página principal.

```
<!--  
    Ejemplo uso $_SESSION[] en login  
-->  
<script src="../../js/login.js"></script>  
<?php
```

```

session_start();
require 'conexion.php';
$nombre=$_POST["alias"];
$pass=$_POST["pass"];

//consulta con la base de datos donde buscamos al usuario
//que ha introducido sus datos en el inicio de sesión
$sql = 'SELECT * FROM Usuarios WHERE nick="'. $nombre. '";';
$resultado = mysqli_query($con, $sql) or die("Error por nick en consulta sobre
la tabla Usuarios");
$usuario = array();
while($fila = mysqli_fetch_array($resultado)){
    $usuario[] = $fila;
}
$passdb=$usuario[0]['password'];
$passIntroducida=str_replace(' ', '', $pass);
//Contraseñas cifradas con md5
if($passdb == md5($passIntroducida)){
    //Si la consulta ha sido correcta, se asocian al usuario
    //las siguientes variables de sesion
    $_SESSION["nick"] = $nombre;//nick de usuario
    $_SESSION["login"] = true;//identifica al usuario como logeado
    $_SESSION["rol"] = $usuario[0]['rol'];//rol del usuario
    $_SESSION["nivel"] = $usuario[0]['nivel'];//nivel de usuario
    $_SESSION["formularioNivel"]=$usuario[0]['nivel'];//nos indica si ha
realizado la prueba inicial
    require 'desconexion.php';
    header('Location: /');
}
else{
    require 'desconexion.php';
    header('Location: /login.php?login=ko');
}
?>

```

4.2. Usuarios logueados

El usuario, en el momento de su registro o identificación, dependiendo de si es la primera vez que accede a la herramienta, comprueba que el entorno es diferente. En primer lugar, desaparece la página principal donde se describe la herramienta y donde se encuentra un formulario para el registro de usuarios. En cambio, accede directamente a la sección de artículos, donde podrá realizar la consulta de los artículos del rol que desee. Se otorga de un filtro para que la búsqueda sea más fácil, pudiendo ordenar los artículos mediante diferentes criterios como son fecha, mayor nivel y menor nivel [\[RF06\]](#). Se presentan los artículos en una misma página, pudiendo desplegar más artículos mediante un botón (mostrar más). Se le permite la **visualización de las creaciones candidatas** a convertirse en artículos y las **modificaciones sugeridas** por otros usuarios para la incorporación como nuevos artículos, a través de la sección **comunidad**. **Se permite votar** a aquellas creaciones/modificaciones que

se correspondan con el rol del usuario identificado y que no sean superiores a su nivel. El usuario también será capaz de realizar búsquedas sobre los contenidos de la web, editar su perfil de usuario y realizar **pruebas de nivel**.

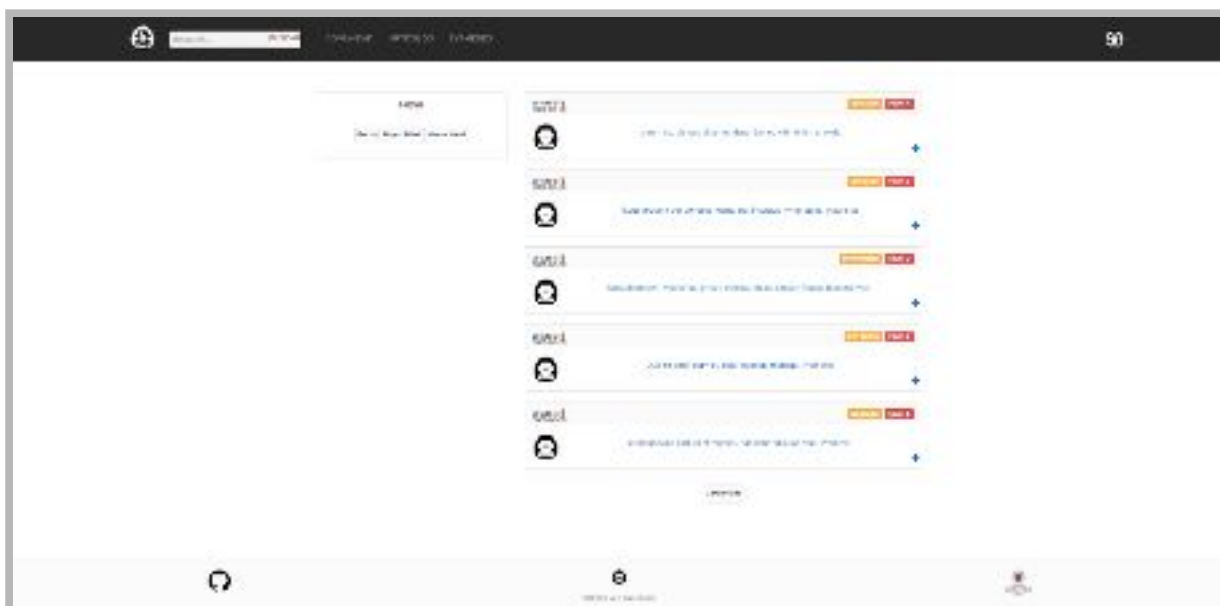


Figura 13. Aspecto de la herramienta para un usuario logueado.

4.2.1. Generación de nuevo contenido

El usuario podrá **generar nuevo contenido** en la sección de **COMUNIDAD->CREAR CONTENIDO** [RF01]. Se le presenta una página donde crear un nuevo artículo. El usuario tendrá la posibilidad de generar un **título**, un **contenido**, **nivel** de la creación y **3 preguntas** como máximo para cada nuevo artículo. También se ofrece la posibilidad de insertar múltiples **imágenes** para cada contribución [RF09]. Deberá rellenar como mínimo los apartados de título, contenido y nivel, pasando a formar parte de las creaciones propuestas por los diferentes usuarios. En caso de no introducir los datos obligatorios se le redigirá a la página de creación mostrándose un mensaje de error.

En la siguiente figura se describe la interfaz para la generación de nuevo contenido. Se puede distinguir un formulario para que el usuario introduzca su contenido y grupos por cada bloque de datos. Título, contenido, preguntas, nivel e imágenes.

```
<form id="formularioArticulo" enctype="multipart/form-data" method="POST"
onsubmit="return updateValue()">
  <hr>
  <div class="form-group">
    <div class="panel-group">
      <div id="titulo" class="panel-body" > Título
      ...
    </div>
  </div>
```

```

<hr>
<div class="form-group"> Contenido
  <div class="panel-group">
    ...
  </div>
</div>
<hr>
<div class="form-group"> Preguntas
  <div class="panel-group">
    ...
  </div>
</div>
<div class="form-group"> Nivel
  <label for="seleccionNivel">Elige el nivel de tu artículo</label>
  <select class="form-control" id="nivel" name="nivel">
    <?php
      for($i=1; $i<=$_SESSION["nivel"];$i++){?>
        <option><?php echo $i ?></option>
      <?php } ?>
    </select>
  </div>
</div>
<hr>
<div class="form-group"> Imagen
  <label for="imagen">Añade imagenes</label>
  <input type="file" class="form-control-file" id="imagen" name="imagen[]"
multiple="multiple">
  ...
  <button class="btn btn-success">Enviar modificación</button>
</form>

```

Comentar como se muestra en la figura, la restricción sobre el nivel de los artículos generados por el usuario como nueva contribución. El nuevo artículo **no podrá ser de mayor nivel que el nivel actual del usuario que lo crea**. [\[RF03\]](#)

Las funcionalidades incorporadas y el almacenamiento en base de datos es el mismo que se describe a continuación en la modificación de contenido.

4.2.2. Modificación de contenido

Se añade la posibilidad de **sugerir modificaciones** sobre artículos ya consolidados. [\[RF01\]](#). El usuario deberá acceder a esta opción haciendo click sobre la cruz existente en cada caja del artículo y acceder a la sección de modificar. En esta página, se ofrece al usuario la posibilidad de editar el título, el contenido y añadir como máximo 3 preguntas. Una vez rellenados dichos apartados, la modificación pasará a ser evaluada por el resto de la comunidad mediante el sistema de votaciones.

Tanto para el proceso de creación de nuevo contenido como para la modificación del ya existente, se intenta dotar al usuario de una interfaz amigable. Esta interfaz es la misma que la descrita en el punto anterior (creación de contenido).

A continuación se describen algunas funcionalidades:

Los bloques de edición de título y contenido se **convierten dinámicamente** de un bloque `<div>` a un `<textarea>` donde el usuario puede escribir su artículo. Para que lo que el usuario ha escrito se vuelva a convertir en un bloque div, se incorpora otro botón, en este caso de Guardar, que realiza la operación inversa. Esto se consigue mediante funciones JavaScript.

```
<a class="editar">
<span id="guardarTitulo" class="label label-success pull-right">Guardar
<span class="glyphicon glyphicon-save"></span>
</span></a>
<a class="editar">
<span id="editarTitulo" class="label label-success pull-right">Editar
<span class="glyphicon glyphicon-pencil"></span>
</span></a>

-- Se asocian las funciones js como eventos a las etiquetas editar, guardar --
<script>
var titulo = document.getElementById("editarTitulo");
var contenido = document.getElementById("editarContenido");
titulo.addEventListener("click", editarTitulo);
contenido.addEventListener("click", editarContenido);
</script>

-- Funciones js --
function divClickedTitulo(area) {
    var divHtml = $(area).html();
    var editableText = $("<textarea id='textoTitulo'/>");
    editableText.val(divHtml);
    $(area).replaceWith(editableText);
    textarea="textoTitulo";
    id="textoTitulo";
    editableText.focus();
    $("#guardarTitulo").click(editableTextBlurredTitulo);
}
function editableTextBlurredTitulo() {
    var e = $("textarea#textoTitulo");
    var valor = e.val();
    e.replaceWith('<div style="padding:10px;" id="titulo">' + valor + '</div>');
}
```

Como ya se ha mencionado, para el bloque de contenido, se lanza el enriquecedor de texto **TinyMCE** para dotar de una mayor personalización de la contribución. [\[RF08\]](#) [\[RNF01\]](#) [\[RNF04\]](#)


```
function divClickedContenido(area) {
    var divHtml = $(area).html();
    var editableText = $('<textarea id='textoContenido' style='height:20%;/;>');
    editableText.val(divHtml);
    $(area).replaceWith(editableText);
    tinymce.init({ selector:'textarea#textoContenido' });
    textarea="textoContenido";
    id="textoContenido";
    editableText.focus();
    $('#guardarContenido').click(editableTextBlurredContenido);
}

function editableTextBlurredContenido() {
    if(tinymce.activeEditor!=null) {
        var html = tinymce.activeEditor.getContent();
        tinymce.remove();
        var e = $('textarea#textoContenido');
        var valor = e.val();
        e.replaceWith('<div style="padding:10px;" id="contenido">' + valor +
        '</div>');
        $("body").off('click');
    }
}
```

Como estos bloques no se pasan a backend a través de un formulario ya que no son <input> sino <div>, se añaden en el momento que se envía el formulario, también a través de una función JavaScript. En esta función también se valida que ni el título ni contenido del artículo no sean vacíos, ya que son datos obligatorios.

```
<script type="text/javascript">
var updateValue = function() {
    var titulo = $('#titulo').html();
    var contenido = $('#contenido').html();
    if( $.trim(titulo) == ""){
        $('#formularioArticulo').attr("action","creararticulo.php?enviar=emptyT");
    }else if($.trim(contenido) == ""){
        $('#formularioArticulo').attr("action","creararticulo.php?enviar=emptyC");
    }else{
        $('#enviar').attr("type","submit");
        $('#formularioArticulo').attr("action","backend/creararticulo.php");
        $('#formularioArticulo').append("<input type='hidden' name='titulo'
value='"+$('#titulo').html()+"'>");
        $('#formularioArticulo').append("<input type='hidden' name='contenido'
value='"+$('#contenido').html()+"'>");
    }
}
```



En el caso de las preguntas, el usuario podrá incluir 3 preguntas como máximo, que se insertarán en campos del formulario y que posteriormente se guardarán en base de datos asociadas a la contribución.

Se podrán introducir múltiples imágenes asociadas a la contribución [\[RF09\]](#). Se guardan en un directorio del servidor identificado con el id de la contribución. Como se ha mencionado, este mecanismo se ha desarrollado para evitar una sobrecarga en la base de datos. Es un sistema que guarda en una ruta del servidor las imágenes identificadas por el id de la contribución como nombre del directorio donde se encuentran numeradas las imágenes asociadas al id de la contribución. En base de datos se guardará la correspondiente relación entre el path y el id de la contribución. En el siguiente código se explica el proceso de guardado de un artículo junto con sus datos y las imágenes.

```
<?php
    session_start();
    require("conexion.php");
    if(isset($_SESSION['login'])){
        if ($_SERVER["REQUEST_METHOD"] == "POST") {
            $id=0;
            require("recogeDatosArticulo.php");
            require("insertaArticulo.php");
            require("subeImgArticulo.php");
        }
    }
    $_SESSION['creacion']=true;
    require("desconexion.php");
    header('Location: /');
?>
```

```
<?php
RecogeDatosArticulo
    $nivel=$_POST["nivel"];
    $rol=$_SESSION["rol"];
    $titulo=$_POST["titulo"];
    $contenido=$_POST["contenido"];
    $creador= $_SESSION['nick'];

    $pregunta0=$_POST["textoPregunta0"];
    $respuestaA0=$_POST["respuestaA0"];
    $respuestaB0=$_POST["respuestaB0"];
    $respuestaC0=$_POST["respuestaC0"];
    $respuestaD0=$_POST["respuestaD0"];
    $correcta0=$_POST["correcta0"];
    ...
?>
```

```
InsertaArticulo
```



```
$sql = "INSERT INTO Modificaciones (votos_positivos, votos_negativos, nick) VALUES
(0, 0 , '$creador');";
    $consulta = mysqli_query($con, $sql) or die("No se pudo insertar la
modificacion");
    $id_mod = mysqli_insert_id($con);

    $sql = "INSERT INTO Modificacion_articulo (id_mod, id, rol, nivel, titulo,
contenido, creador) VALUES ('$id_mod', '$id', '$rol', '$nivel', '$titulo',
'$contenido', '$creador');";
    $consulta = mysqli_query($con, $sql) or die("No se pudo insertar el articulo");

    if ($pregunta0!=""){
        $sql = "INSERT INTO Modificacion_pregunta (rol, id_mod, nivel, texto, A, B,
C, D, Correcta, id) VALUES ('$rol', '$id_mod', '$nivel', '$pregunta0',
'$respuestaA0', '$respuestaB0', '$respuestaC0', '$respuestaD0', '$correcta0',
'$id');";
        $consulta = mysqli_query($con, $sql) or die("No se pudo insertar la
pregunta0");
    }
    if ($pregunta1!=""){
        ...
    }
    if ($pregunta2!=""){
        ...
    }
}

<?php SubeImgArticulo
    $total = count($_FILES['imagen']['name']);
    for($i=0; $i<$total; $i++) {
        $ficheroAsubir = $_FILES['imagen']['tmp_name'][$i];
        $directorio = "../files/img/articulos/" . $id_mod . "/" ;
        $nombreImagen = $_FILES['imagen']['name'][$i];
        $fichero_subido = $directorio . $nombreImagen;
        if (!file_exists($directorio)) {
            mkdir($directorio, 0777, true); //crea el directorio si no existe
        }
        if (move_uploaded_file($ficheroAsubir, $fichero_subido)) {
            echo "El fichero es válido y se subió con éxito.\n";
            $sql = "INSERT INTO Imagen_modificacion_articulo (id_mod, path) VALUES
('$id_mod', '$nombreImagen');";
            $consulta = mysqli_query($con, $sql) or die("No se pudo insertar la
imagen");
        } else {
            echo "¡Posible ataque de subida de ficheros!\n";
        }
        echo $fichero_subido;
    }
}
```

4.2.3. Votaciones sobre contribuciones de usuarios

El sistema está preparado para la incorporación de nuevos artículos sin la necesidad de tener un administrador dedicado a insertar nuevo contenido a la web [\[RF04\]](#). Esto es posible gracias a la oportunidad que tienen los usuarios de votar las contribuciones que se encuentran en la sección **COMUNIDAD**.

Cada contribución tendrá una valoración de votos positivos y votos negativos. En el momento que un usuario accede a una contribución **de su mismo rol**, tendrá la posibilidad de **votar** dicha contribución al igual que añadir un comentario a su votación (tanto en el caso de una nueva creación de artículo o una modificación de un artículo existente). Estos datos serán almacenados en base de datos para la restricción de un sólo voto por usuario como se explica más adelante.

En el momento que una contribución tenga un número total de votos pasará a ser evaluada. Si el porcentaje de votos positivos cumple con el criterio consensuado previamente por los administradores, se convierte automáticamente en un nuevo artículo.

El proceso se define a grandes rasgos a continuación. En caso de cumplir con el criterio definido debemos distinguir si la contribución pertenece a un artículo previo (modificación), o bien es una contribución referente a un nuevo artículo (creación). En el caso de ser una modificación, en el momento que esta se consolide como un nuevo artículo se deberá eliminar el artículo sobre el que se ha realizado dicha modificación. También deberán eliminarse las preguntas asociadas a dicho artículo que se encuentran en la tabla preguntas.

En ambos casos se debe eliminar la tupla correspondiente en las tablas de Modificaciones_artículo y Modificaciones_pregunta. De esta manera se evita una sobrecarga en la base de datos y evitamos duplicidad en nuestros datos.

El procedimiento de consolidación de las contribuciones se puede ver en el código correspondiente a **votos.php**. En caso de que el voto introducido sea positivo, entramos a valorar si la contribución cumple con el criterio establecido. Debe de tener un porcentaje de votos positivos mayor del sesenta por ciento y además el número total de votos debe ser mayor a 100.

En el caso de cumplir dicho criterio se insertan en las tablas Artículos y Preguntas las tuplas correspondientes a la modificación (se encuentran en las tablas Modificaciones_artículos y Modificaciones_preguntas). Posteriormente se comprueba si la contribución pertenece a un artículo previo en la aplicación y, en ese caso, se eliminan las tuplas de las tablas Artículos y Preguntas. Posteriormente se eliminan las tuplas de la propia Modificación para evitar sobrecargar la base de datos con contribuciones consolidadas como artículos.

```
CÓDIGO CORRESPONDIENTE A votos.php
<?php
    <?php
        session_start();
        require 'conexion.php';
        $voto=$_POST['voto'];
        $contribucion=$_POST['contribucion'];
        $nick=$_POST['usuario'];
        $comentario=$_POST['comentario'];
        $sql = 'SELECT * FROM Modificaciones WHERE id_modificacion LIKE
        "'. $contribucion. '"';
        $result=mysqli_query($con, $sql);
        if ($result){
            $fila = mysqli_fetch_array($result);
            $votos[] = $fila;
        }else{
            echo ('La consulta de votaciones usuario ha fallado');
        }

        //Si el voto es negativo no entramos a insertar en articulos consolidados
        if ($voto=="negativo"){
            $sqlInsert = "UPDATE Modificaciones SET votos_negativos=votos_negativos+1
        WHERE id_modificacion = '$contribucion'";
        }else{

            $sqlInsert = "UPDATE Modificaciones SET votos_positivos=votos_positivos+1
        WHERE id_modificacion = '$contribucion'";
            $positivos=$votos[0]['votos_positivos'];
            $negativos=$votos[0]['votos_negativos'];
            $total=$positivos+$negativos;
            $porcentaje=($positivos*100)/$total;

            //Actualizacion de la creacion en articulos consolidados
            if ($porcentaje > 60 && $total>1){
                //Seleccionamos los datos de la modificacion para insertar en articulos
                como consolidado
                $sql = 'SELECT * FROM Modificacion_articulo WHERE id_mod LIKE
                "'. $contribucion. '"';
                $result=mysqli_query($con, $sql);
                if ($result){
                    $filaMod = mysqli_fetch_array($result);
                    $modificacion[] = $filaMod;
                    //Insertamos en articulos
                    $nid=$modificacion[0]['id_mod'];
                    $nnivel=$modificacion[0]['nivel'];
```

```

        $nrol=$modificacion[0]['rol'];
        $ntitulo=$modificacion[0]['titulo'];
        $ncontenido=$modificacion[0]['contenido'];
        $ncreador=$modificacion[0]['creador'];
        $nfecha=$modificacion[0]['fecha'];
        $InsertArticulos = "INSERT INTO Articulos (id, nivel, rol, titulo,
contenido, creador, fecha) VALUES ('$nid', '$nnivel', '$nrol', '$ntitulo',
'$ncontenido', '$ncreador', '$nfecha');"
        $consulta1 = mysqli_query($con, $InsertArticulos) or die("Error en
consulta sobre la tabla arti, ". mysqli_error($con) . " " . $InsertArticulos);

        //Seleccionamos las preguntas de la contribucion y las insertamos en
preguntas consolidadas
        $sqlPreguntas = 'SELECT * FROM Modificacion_pregunta WHERE id_mod
LIKE "'. $contribucion.'"';
        $resultPreguntas=mysqli_query($con, $sqlPreguntas);
        if ($resultPreguntas){
            while($row = mysqli_fetch_assoc($resultPreguntas)){
                $key = md5(microtime().rand());
                $pid=$row['id_mod'];
                $pnivel=$row['nivel'];
                $ptexto=$row['texto'];
                $prol=$row['rol'];
                $pA=$row['A'];
                $pB=$row['B'];
                $pC=$row['C'];
                $pD=$row['D'];
                $pCorrecta=$row['correcta'];
                $pId=$row['nivel'];
                $InsertPreguntas = "INSERT INTO Preguntas (id_pregunta,
nivel, texto, rol, A, B, C, D, correcta, id) VALUES ('$key', '$pnivel', '$ptexto',
'$prol', '$pA', '$pB', '$pC', '$pD', '$pCorrecta', '$pid');"
                $consulta2 = mysqli_query($con, $InsertPreguntas) or
die("Error en consulta sobre la tabla preg, ". mysqli_error($con) . " " .
$InsertPreguntas);
            }
        }

        //caso de ser una modificacion de articulo existente borramos el
articulo anterior y sus preguntas
        if($modificacion[0]['id'] != '0'){
            $sqlDelete="DELETE FROM Articulos WHERE
id='". $modificacion[0]['id']."'";
            mysqli_query($con, $sqlDelete) or die(mysqli_error($con));

            $sqlDelete="DELETE FROM Preguntas WHERE
id_pregunta='". $modificacion[0]['id']."'";

```



```
        mysqli_query($con, $sqlDelete) or die("Error en consulta sobre
la tabla Preguntas, ". mysqli_error($con) . " " . $sqlDelete);
    }
    //BORRAR DE MODIFICACION_PREGUNTA Y MODIFICACION_ARTICULO
    $sqlDelete="DELETE FROM Modificacion_articulo WHERE
id_mod='".$contribucion.'";";
    mysqli_query($con, $sqlDelete) or die("Error en consulta sobre la
tabla modArticulos, ". mysqli_error($con) . " " . $sqlDelete);

    $sqlDelete="DELETE FROM Modificacion_pregunta WHERE
id_mod='".$contribucion.'";";
    mysqli_query($con, $sqlDelete) or die("Error en consulta sobre la
tabla modArticulos, ". mysqli_error($con) . " " . $sqlDelete);
    }else{
        echo ('La consulta de votaciones usuario ha fallado');
    }
}
}
if (mysqli_query($con, $sqlInsert)) {
    echo 'Todo ha ido bien';
}else{
    echo 'La consulta ha fallado';
}
//Insertamos en la tabla VotacionesContribuciones para restriccion usuario vote
1 sola vez
$sqlContribu = "INSERT INTO Votaciones_Contribuciones(contribucion, nick, voto,
comentario) VALUES ('$contribucion','$nick','$voto','$comentario')";
if (mysqli_query($con, $sqlContribu)) {
    header('Location:/comunidad.php');
}else{
    echo 'La consulta ha fallado';
}
include 'desconexion.php';
?>
```

Por último, tanto si la contribución cumple el criterio de consolidación como si no, se crea una nueva tupla en la tabla Votaciones_contribuciones. Esta tabla nos servirá para que un usuario no pueda votar la contribución en más de una ocasión. Esto se consigue almacenando en dicha tabla el id de la contribución, el nick del usuario que ha votado, la valoración del voto y por último, un comentario asociado a dicho voto.

Al entrar el usuario en una contribución se comprueba si existe una tupla en esta tabla con dicho usuario y el id de dicha contribución. En ese caso no se pintará el formulario de envío de votos como se puede comprobar en el código correspondiente a la restricción de votos sino que se le mostrará un mensaje que indique qué tipo de valoración ha hecho el usuario sobre esta contribución.

```

CÓDIGO CORRESPONDIENTE A LA RESTRICCIÓN DE VOTOS
session_start();
include '/backend/conexion.php';
$id=$_GET["id"];
$sql = 'SELECT * FROM Votaciones_Contribuciones WHERE nick LIKE
"'.$_SESSION['nick'].'"' AND contribucion LIKE "'.$id.'"';
$result=mysqli_query($con, $sql);
if ($result){
    $fila = mysqli_fetch_array($result);
    $votaciones[] = $fila;
    $numRegistros = mysqli_num_rows($result);
}else{
    echo ('La consulta de votaciones usuario ha fallado');
}
...

<?php
    if(isset($_SESSION['login']) && $_SESSION['rol']==$rol && $_SESSION['nivel']
> $nivel && $numRegistros==0){
        ?>

        <div class="caja votos col-xs-12 col-sm-6 col-sm-offset-3 col-md-6
col-md-offset-3 col-lg-6 col-lg-offset-3">
            <form id="votarContribucion" method="POST" action="/backend/votos.php"
onsubmit="return completarDatos()">
                <div id="votacion" class="form-group">
                    ...
                </div>
            </form>
        </div>
    <?php }
    else{
        if($votaciones[0]['voto']=="positivo"){ ?>
            <div id="alert success" class="alert alert-success" role="alert">Te ha
gustado esta contribución</div>
            <?php }
            else{ ?>
                <div id="alert false" class="alert alert-danger" role="alert">No te ha
gustado esta contribución</div>
                <?php }
            } ?>

```


4.2.4. Pruebas de nivel

El usuario tiene la posibilidad de mejorar su nivel de autodefensa digital mediante la realización de diferentes pruebas de nivel. Esta funcionalidad se encuentra en el apartado **PRUEBAS DE NIVEL** [\[RF07\]](#). En esta página se le presentan al usuario sus pruebas realizadas, aquellas que ha superado se identifican con el color verde, mientras que la prueba correspondiente al siguiente nivel a alcanzar por el usuario se identifica mediante el color azul.

El usuario sólo tendrá disponible acceder al examen de un nivel superior al suyo actual, esto se consigue mediante una validación en el lado del navegador de la siguiente manera.

```
<a <?php if(!isset($_SESSION['login'])) {?>
    class="list-group-item disabled" <?php
} else { ?> class="list-group-item
    <?php if($i<=$_SESSION['nivel']) { ?>
        list-group-item-success"<?php
    } else { ?> list-group-item-info" href="/examen.php?id=<?php echo $i ?>"<?php
} }?>>
```

Siendo \$i un contador que se incrementa por cada nivel y que se utiliza para determinar el formato de cada elemento de la lista (prueba de nivel). Si el usuario no está logueado, será imposible entrar, mientras que si está logueado sólo tendrá acceso a su nivel inmediatamente superior.

Al entrar en la prueba, el usuario se encuentra con una serie de preguntas relacionadas con el nivel y su rol, debiendo contestar correctamente para subir de nivel.

Para la gestión de las **pruebas de nivel**, se ha utilizado un **formato** de texto **JSON**. Cada nivel tiene un fichero JSON asociado, de manera que cuando el usuario quiere realizar una prueba, se parsea dicho fichero y se muestran las preguntas para la evaluación. Las respuestas de la prueba se validan en la parte back de la aplicación, recogiendo las opciones marcadas por el usuario y comprobando con la respuesta correcta de cada pregunta. La respuesta correcta se obtiene del mismo fichero JSON que se vuelve a recorrer por cada pregunta.

```
{
  "p1": {
    "texto": "Como cifrar el correo",
    "correcta": "A",
    "puntos": 2,
    "respuestas": [
      {
        "texto": "A)",
        "valor": "A"
      },
      {
```

```

        "texto": "B)",
        "valor": "B"
    },
    {
        "texto": "C)",
        "valor": "C"
    },
    {
        "texto": "D)",
        "valor": "D"
    }
]
},
"p2": {
    "texto": "Montar una VPN",
    "correcta": "A",
    ...
}

```

Por otro lado, desde el lado del backend, se recogen los datos de las respuestas de los usuarios a través del formulario de examen y se comprueban con los valores correctos de las preguntas, presentes en el fichero .json

```

<?php
    session_start();
    require 'conexion.php';
    $niv=$_SESSION['nivel']+1;
    $jsonText = file_get_contents("../doc/examenNivel".$niv.".json");
    $preguntas = json_decode($jsonText, true);
    $puntos=0;
    $nombre=$_SESSION['nick'];
    foreach($_POST as $key => $value){
        if($preguntas[$key]['correcta'] == $value){
            $puntos+=$preguntas[$key]['puntos'];
        }
    }
    if($puntos > 5){
        $sqlInsert = "UPDATE Usuarios SET nivel = '$niv' WHERE nick = '$nombre'";
        if (mysqli_query($con, $sqlInsert)) {
            $sql = "SELECT * FROM Usuarios WHERE nick='".$nombre."'";
            $resultado = mysqli_query($con, $sql) or die("Error por nick en consulta sobre la tabla Usuarios");
            $usuario = array();
            $_SESSION["formularioNivel"] = true;
            $_SESSION["nivel"] = $niv;

```

```
        $_SESSION["examen"]='aprobado';
    } else {
        echo("Error en la consulta");
    }
} else {
    $_SESSION["examen"]='suspense';
}
include 'desconexion.php';
header('Location: /');
?>
```

4.2.5. Perfil del usuario

La aplicación permite al usuario tener una sección dedicada al mismo. Se le presentan sus datos correspondientes al **rol**, **nivel**, **imagen** de perfil y **email** [\[RF10\]](#). En esta sección el usuario tiene la posibilidad de **consultar** sus **contribuciones** a la herramienta, tanto para sus nuevas creaciones como para las modificaciones sobre artículos existentes. También tendrá disponibles sus contribuciones que han pasado a formar parte de artículos consolidados en la herramienta.

Esto se consigue mediante consultas a la base de datos, buscando aquellas contribuciones que sean del usuario almacenado en la variable de sesión **nick** del usuario. La diferencia entre una creación y una modificación reside en el resultado de la consulta. Ambas atacan a la misma tabla de la base de datos, pero en el caso de ser una modificación, dispondrá de valor en el campo id referente a un artículo consolidado de la aplicación. Mientras que una contribución referente a un nuevo artículo, no almacenará ningún valor en el campo id de la base de datos.

```
<?php
if(isset($_GET["id"])&&$_GET["id"]!=""){
    $sql ='SELECT * FROM Modificacion_articulo a, Modificaciones m WHERE a.id <> 0
AND a.id_mod = m.id_modificacion AND a.creador = "'. $_GET["id"] .'" ORDER BY fecha
DESC;';
?>
```

En este ejemplo podemos ver cómo se consulta sobre modificaciones_artículo filtrando los resultados a través de la condición de que el id sea distinto de cero. De esta manera, obtendremos sólo las modificaciones de artículos ya existentes. El cruce con la tabla de modificaciones es para sacar los votos, ya que al estar diseñado el modelo mediante herencia, la tabla modificaciones comparte sus campos con modificaciones_articulo y modificaciones_preguntas. Es en estas tablas donde se guardan las contribuciones de los usuarios, tanto la información referente al contenido de la contribución como las preguntas asociadas a la misma.



El usuario puede acceder a su zona de edición de perfil donde **podrá modificar sus datos personales** como **email**, **contraseña** e **imagen** de perfil. En el caso de la imagen, conviene señalar que el usuario, al registrarse en la aplicación se le asocia una imagen por defecto. La zona de modificación de datos de perfil ofrece la posibilidad de sustituir esta imagen por defecto. Internamente el proceso consiste en sustituir el archivo en el servidor de la aplicación. El nombre de dicho archivo se corresponde con el nick del usuario en la aplicación.

La edición del perfil realiza varias validaciones correspondientes a los datos que el usuario ha introducido para su edición. En primer lugar se comprobará que la contraseña introducida como clave para la edición de los datos personales sea la misma que la que tiene el usuario en la base de datos. A continuación se comprueba si el usuario ha introducido el email. Esto se realiza porque el email no es un campo obligatorio y la sentencia de actualización es diferente para cada caso. En caso de insertarse, se comprobará que no exista actualmente ninguno en la base de datos, y posteriormente se reemplaza la nueva imagen por la que existe en el servidor asociada al usuario. En caso de no insertarse el email, se actualizan los datos de la contraseña y se reemplaza la imagen en caso de haber subido alguna. Si durante el proceso alguna de las validaciones fallan, se redirigirá a la zona de edición de perfil y se mostrará un mensaje de excepción que lo determinará una variable de tipo \$_GET[].

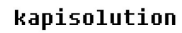
```
<?php
session_start();
require 'conexion.php';

$emailUso=false;
$email=$_POST['email'];
$passant=$_POST['passant'];
$pass=$_POST['pass'];
$passrep=$_POST['pass-repeat'];
$id=$_SESSION['nick'];

$sql1 = 'SELECT password FROM Usuarios WHERE nick LIKE "'. $id. '";';
$passanterior = mysqli_query($con, $sql1) or die("Error por pass en consulta sobre
la tabla Usuarios");
$row = mysqli_fetch_assoc($passanterior);
$passComprob = $row['password'];

if($pass==$passrep && $passComprob==md5($passant)){
    $passdb = md5($pass);
    if($email!=''){
        $sqlEmail = 'SELECT * FROM Usuarios WHERE email LIKE "'. $email. '";';
        $resultadoEmail = mysqli_query($con, $sqlEmail) or die("Error por email en
consulta sobre la tabla Usuarios");
        $numRegistrosEmail = mysqli_num_rows($resultadoEmail);

        if($numRegistrosEmail>0){
            $emailUso=true;
            header('Location: /editarusuario.php?koem');
```



```
//Validaciones en parte frontal para mostrar mensaje de error
if(isset($_GET['kom'])){?>
    <div class="alert alert-danger text-center" role="alert">Fallo en la
edición del perfil. Recuerda introducir correctamente tu contraseña actual</div>
<?php }
    else if(isset($_GET['koem'])){?>
    <div class="alert alert-danger text-center" role="alert">Fallo en la
edición del perfil. Email ya en uso</div>
<?php } ?>
```

4.2.6. Búsqueda

La herramienta ofrece la posibilidad de realizar búsquedas a través de la barra de navegación [\[RF05\]](#).

El proceso consiste en la consulta a la base de datos de la cadena introducida a través del formulario de búsqueda de la barra de navegación. Los resultados se distinguen entre usuarios, artículos, creaciones de artículos y modificaciones de artículos. En las siguientes figuras se muestran tanto la consulta con la base de datos como la presentación de los distintos resultados.

```
<?php
if(isset($_GET["search"])&&$_GET["search"]!=""){
    $sql = 'SELECT * FROM Usuarios WHERE nick LIKE "%'.$_GET["search"] .'%";';
    $consulta = mysqli_query($con, $sql) or die("Error en consulta sobre la tabla
usuarios");
    $busquedaUsuarios = array();
    while($fila = mysqli_fetch_array($consulta)){
        $busquedaUsuarios[] = $fila;
    }
}

if(isset($_GET["search"])&&$_GET["search"]!=""){
    $sql = 'SELECT * FROM Articulos WHERE titulo LIKE "%'.$_GET["search"] .'" OR
contenido LIKE "%'.$_GET["search"] .'" ORDER BY nivel, rol;';
    $consulta = mysqli_query($con, $sql) or die("Error en consulta sobre la tabla
articulos");
    $busquedaArticulos = array();
    ...
}

if(isset($_GET["search"])&&$_GET["search"]!=""){
    $sql = 'SELECT * FROM Modificacion_articulo a, Modificaciones m WHERE a.id = 0
AND a.id_mod = m.id_modificacion AND a.titulo LIKE "%'.$_GET["search"] .'" AND
a.contenido LIKE "%'.$_GET["search"] .'" ORDER BY a.nivel, a.rol;';
    $consulta = mysqli_query($con, $sql) or die("Error en consulta sobre la tabla
creacion de articulos");
    $busquedaCreaciones = array();
    ...
}

if(isset($_GET["search"])&&$_GET["search"]!=""){
    $sql = 'SELECT * FROM Modificacion_articulo a, Modificaciones m WHERE a.id <> 0
AND a.id_mod = m.id_modificacion AND a.titulo LIKE "%'.$_GET["search"] .'" AND
a.contenido LIKE "%'.$_GET["search"] .'" ORDER BY a.nivel, a.rol;';
    $consulta = mysqli_query($con, $sql) or die("Error en consulta sobre la tabla
modificaciones de articulos");
}
```

```

$busquedaModificaciones = array();
...
}
?>

```

```

<?php
    require 'backend/busqueda.php';
    $contador=0;
?>
<div class="container">
    <hr>
    <div class="row">
        <div class="col-xs-12 col-sm-12 col-md-4 col-lg-4">
            <div class="list-group">
                <button id="usr" type="button" class="list-group-item disabled"
onclick="mostrarTarjetas(event, 'albumUsuarios', 'usr')">
                    Usuarios
                </button>
                <button id="art" type="button" class="list-group-item"
onclick="mostrarTarjetas(event, 'articulos', 'art')">Articulos</button>
                <button id="crea" type="button" class="list-group-item"
onclick="mostrarTarjetas(event, 'creaciones', 'crea')">Creaciones</button>
                <button id="mod" type="button" class="list-group-item"
onclick="mostrarTarjetas(event, 'modificaciones', 'mod')">Modificaciones</button>
            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-8 col-lg-8">
            <!--BusquedaUsuarios-->
            <div id="albumUsuarios" class="album py-5 bg-light">
                ...
            </div>
            <!--BusquedaArticulos-->
            <div class="album py-5 bg-light" id="articulos" style="display:none">
                ...
            </div>
            <!--busquedaCreaciones-->
            <div class="album py-5 bg-light" id="creaciones" style="display:none">
                ...
            </div>
            <!--BusquedaModificaciones-->
            <div class="album py-5 bg-light" id="modificaciones"
style="display:none">
                ...
            </div>
        </div>
    </div>

```

```
</div>
```

La función `mostrarTarjetas` es una función JavaScript encargada de mostrar el bloque en cuestión (usuarios, artículos,...) que se le pasa a la función a través de su cabecera.

```
function mostrarTarjetas(evt, tarjetero, btn) {  
    var x;  
    x = document.getElementsByClassName("album py-5 bg-light");  
    for (i = 0; i < x.length; i++) {  
        x[i].style.display = "none";  
    }  
    x = document.getElementsByClassName("list-group-item disabled");  
    for (i = 0; i < x.length; i++) {  
        x[i].className='list-group-item';  
    }  
    document.getElementById(tarjetero).style.display = "block";  
    $("#"+btn).removeClass('list-group-item').addClass('list-group-item disabled');  
}
```

5. Evolución del proyecto

Los objetivos del proyecto han estado definidos desde el primer día hasta el último. Si bien es cierto que el modelo de desarrollo no estaba definido, el deseo era desarrollar una herramienta colaborativa donde el objetivo principal era dotar a diferentes perfiles profesionales de una herramienta donde se localizarán fácilmente artículos con una finalidad común, el conocimiento entorno a la autodefensa digital.

En cuanto al desarrollo, el proyecto ha pasado por una serie de etapas. Este apartado pretende definir a grandes rasgos las líneas de desarrollo seguidas.

En un primer lugar se tomó la decisión de realizar una herramienta desde cero, esto es, una herramienta web sin la incorporación de ningún framework ni librería JavaScript. Preferimos dar un enfoque más útil que bonito a la aplicación, desarrollando funcionalidades de la herramienta en paralelo cada integrante del proyecto.

Posteriormente, para definir una interfaz amigable para el usuario, se tomó la decisión de incluir un framework de desarrollo web (bootstrap) para realizar el desarrollo responsive de la aplicación, otorgando valor así valor a la herramienta y ayudando en las tareas de maquetado de la web.

Finalmente, el uso de este framework ha permitido el desarrollo de la parte frontal de la web de manera conjunta con el desarrollo de las diferentes funcionalidades de la herramienta.

6. Conclusiones

En primer lugar, comentar la satisfacción de haber cumplido los objetivos establecidos durante la definición del proyecto. El objetivo era dotar al usuario de una plataforma web donde esté comprendida diferente información acerca del ámbito de la autodefensa. Este es un campo muy amplio, y por ello, la búsqueda de información acerca de este tipo de conocimiento es muy complicada. La herramienta tendría más valor en tanto pueda reunir en un mismo lugar conocimiento acerca de este ámbito y ofrecerlo de una manera “personalizada”.

En **KapiSolution** (que es el nombre que hemos asignado a nuestra herramienta), un usuario puede encontrar recogidos artículos con diferente contenido pero la misma motivación. Además puede ir aumentando su conocimiento a lo largo del tiempo, y puede recibir un feedback del mismo, a través de las pruebas de nivel. La función colaborativa ha sido un punto importante durante todo el desarrollo, a parte de dotar al usuario de un entorno fácil de edición se ha integrado con una tecnología como TinyMCE, se brinda la oportunidad de votar y comentar cada contribución, de tal manera que son los propios usuarios los que deciden sobre el contenido de la web.

Destacar también la incorporación del framework Bootstrap al proyecto, que permite acceder a nuestra aplicación desde cualquier dispositivo. Aunque el estudio de este framework ha sido amplio y complicado en ocasiones durante la integración, pienso que el resultado final ha merecido la pena.

Además, a la hora de desarrollar el proyecto nos planteamos una serie de requisitos que debía cumplir la herramienta y, como se ha explicado en esta memoria, dichos requisitos han sido conseguidos. También es importante destacar la realización del proyecto bajo tecnologías actuales, permitiendo al usuario interactuar con una herramienta familiar por un lado, y por el lado del desarrollador, ser accesible en el objetivo de crear una herramienta colaborativa, donde pueda ir creciendo a lo largo del tiempo gracias al repositorio de GitHub. Otro aspecto a destacar es que toda la tecnología utiliza es Open Source y la licencia de la herramienta es GPL.

La curva de aprendizaje ha sido difícil, durante la realización del Trabajo Fin de Grado he tenido que familiarizarme con una nueva forma de desarrollo como ha sido el caso del stack LAMP. El uso de lenguajes de programación cuyos conocimientos no eran los requeridos para desarrollar una aplicación como la que se propuso inicialmente. Destacar los conocimientos adquiridos en el desarrollo bajo el control de versiones de Git, una herramienta cada vez más usada y de gran utilidad en el ámbito laboral. Los problemas han sido variados, pero a través de la metodología de trabajo usada, hemos aprendido a colaborar en cualquier desarrollo de software libre a través de una herramienta de repositorios como puede ser GitHub.

7. Trabajo futuro

Se pueden proponer varios trabajos aplicados a la herramienta actual. Una posible mejora puede ser implementar un sistema de creación automático de exámenes mediante consultas a las preguntas de los artículos de la propia aplicación. De esta manera no necesitarían los exámenes el mantenimiento por parte de los administradores.

Un posible trabajo futuro podría ser integrar la herramienta con un bot que respondiera ante preguntas sobre la autodefensa digital. Aprovechar la información existente en la aplicación y llevarlo a otro tipo de herramienta con una funcionalidad común.

8. Enlaces y Bibliografía

8.1. Enlaces

- [1] <https://github.com/kapisolution/>
- [2] <https://en.wikipedia.org/wiki/Wikipedia>
- [3] <https://autodefensa.softcatala.cat/>
- [4] <https://sec.eff.org/topics>
- [5] <https://github.com/>
- [6] <https://kapisolution.github.io/>
- [7] <https://www.tinymce.com/>
- [8] <https://getbootstrap.com/docs/3.3/>
- [9] https://github.com/kapisolution/kapi_solution/projects/1

8.2. Bibliografía

8.2.1. Wikipedia

- [10] Wikipedia. <https://es.wikipedia.org/wiki/Wikipedia>
- [11] Cómo colaborar en Wikipedia. https://es.wikipedia.org/wiki/Ayuda:C%C3%B3mo_puedes_colaborar
- [14] WordPress. <https://es.wikipedia.org/wiki/WordPress>
- [15] Moodle. <https://es.wikipedia.org/wiki/Moodle>
- [16] Wiki Media. https://es.wikipedia.org/wiki/Fundaci%C3%B3n_Wikimedia
- [17] Stack LAMP. https://en.wikipedia.org/wiki/LAMP_%28software_bundle%29
- [19] Apache. https://es.wikipedia.org/wiki/Servidor_HTTP_Apache
- [20] MySQL. <https://es.wikipedia.org/wiki/MySQL>
- [21] PHP. <https://es.wikipedia.org/wiki/PHP>
- [22] HTML5. <https://es.wikipedia.org/wiki/HTML>
- [23] CSS. https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada
- [24] JavaScript. <https://es.wikipedia.org/wiki/JavaScript>

8.2.2. Guía de autodefensa catalana

- [12] <https://autodefensa.softcatala.cat/>

8.2.3. Security Education Companion

- [13] <https://sec.eff.org/>

8.2.4. Bootstrap

- [25] Bootstrap <https://getbootstrap.com/docs/3.3/>

8.2.5. Git

- [18] Control de versiones de Git.

<https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Acerca-del-Control-de-Versiones>

Apéndice

Casos de uso

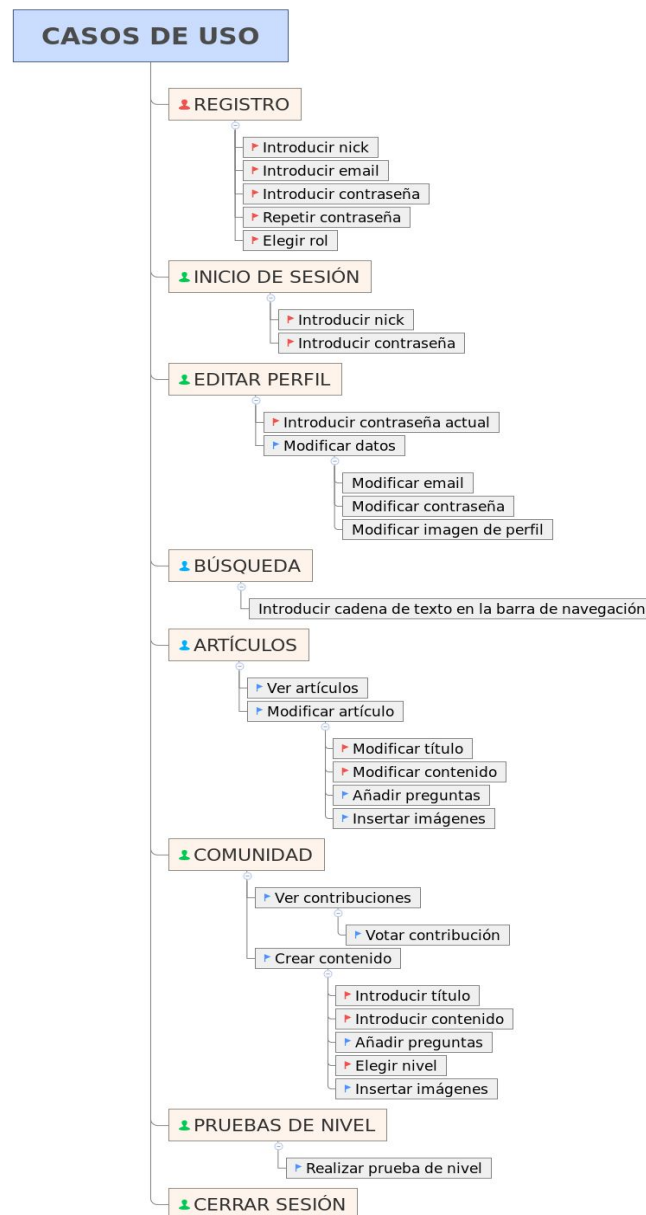


Figura 14. Casos de uso de la aplicación

CU01	Registro
Requisitos	1. Usuario no registrado
Descripción	Un usuario se registra en la herramienta introduciendo los datos necesarios para dicho registro.
Pasos	<ol style="list-style-type: none"> 1. Introducir nick 2. Introducir email 3. Introducir contraseña 4. Repetir contraseña 5. Elegir rol
Sección	REGISTRO
Excepciones	1. Usuario no registrado

CU02	Inicio de sesión
Requisitos	1. Usuario registrado
Descripción	Un usuario registrado inicia sesión en la herramienta introduciendo sus datos..
Pasos	<ol style="list-style-type: none"> 1. Introducir nick 2. Introducir contraseña
Sección	REGISTRO
Excepciones	1. Usuario no registrado

CU03	Editar perfil
Requisitos	<ol style="list-style-type: none"> 1. Usuario registrado. 2. Introducir su contraseña actual.
Descripción	El usuario puede modificar sus datos actuales en el sistema como son: contraseña, email e imagen de usuario.
Pasos	<ol style="list-style-type: none"> 1. Introducir contraseña actual. 2. Modificar datos.
Sección	PERFIL DE USUARIO
Excepciones	<ol style="list-style-type: none"> 1. Usuario no registrado. 2. La contraseña introducida como la actual no se corresponde con la existente en el sistema.

CU04	Búsqueda
Requisitos	Ninguna
Descripción	El usuario puede realizar búsquedas en el sistema correspondientes a usuarios, artículos, creaciones y modificaciones (contribuciones).
Pasos	<ol style="list-style-type: none"> 1. Introducir cadena de texto en la barra de navegación
Sección	GENERAL
Excepciones	Ninguna

CU05	Ver artículos
Requisitos	Ninguna
Descripción	El usuario tiene la posibilidad de ver los artículos disponibles en el sistema.
Pasos	<ol style="list-style-type: none"> 1. Acceder a la sección de artículos. 2. Acceder al artículo deseado.
Sección	ARTÍCULOS
Excepciones	Ninguna

CU06	Modificar artículo
Requisitos	<ol style="list-style-type: none"> 1. Usuario registrado 2. Contenido de nivel inferior o igual al del usuario. 3. Contenido del mismo rol que el del usuario.
Descripción	El usuario modifica un artículo pasando a formar parte de las nuevas contribuciones y con ello, a la votación por parte del resto de usuarios.
Pasos	<ol style="list-style-type: none"> 1. Modificar título. 2. Modificar contenido. 3. Añadir preguntas.. 4. Añadir imágenes a la contribución.
Sección	ARTÍCULOS
Excepciones	<ol style="list-style-type: none"> 1. Usuario no registrado. 2. Usuario con nivel superior. 3. Usuario con distinto rol.

CU07	Ver contribuciones
Requisitos	1. Usuario registrado
Descripción	El usuario accede a las contribuciones creadas por la comunidad (creaciones, modificaciones).
Pasos	1. Acceder a creaciones o modificaciones.
Sección	COMUNIDAD
Excepciones	1. Usuario no registrado

CU08	Votar contribuciones
Requisitos	1. Usuario registrado 2. Contenido del mismo rol que el del usuario.
Descripción	El usuario vota una contribución de manera positiva o negativa. Puede añadir opcionalmente un comentario.
Pasos	1. Votar contribución. 2. Añadir comentario.
Sección	COMUNIDAD
Excepciones	Ninguna

CU09	Crear contenido
Requisitos	<ol style="list-style-type: none"> 1. Usuario registrado 2. Contenido de nivel inferior o igual al del usuario
Descripción	El usuario crea un nuevo artículo pasando a formar parte de las nuevas contribuciones y con ello, a la votación por parte del resto de usuarios.
Pasos	<ol style="list-style-type: none"> 1. Introducir título. 2. Introducir contenido. 3. Añadir preguntas. 4. Elegir nivel. 5. Insertar imágenes a la contribución.
Sección	COMUNIDAD
Excepciones	<ol style="list-style-type: none"> 1. Usuario no registrado

CU10	Realizar pruebas de nivel
Requisitos	<ol style="list-style-type: none"> 1. Usuario registrado. 2. Prueba de un nivel superior al actual.
Descripción	El usuario tiene la posibilidad de aumentar su nivel a través de las pruebas de nivel disponibles en el sistema.
Pasos	<ol style="list-style-type: none"> 1. Acceder a la sección pruebas de nivel. 2. Realizar prueba de nivel.
Sección	PRUEBAS DE NIVEL
Excepciones	<ol style="list-style-type: none"> 1. Usuario no registrado.